

# Comparing apples and oranges: measuring differences between exploratory data mining results

Nikolaj Tatti · Jilles Vreeken

Received: 27 October 2011 / Accepted: 5 June 2012 / Published online: 22 June 2012  
© The Author(s) 2012

**Abstract** Deciding whether the results of two different mining algorithms provide significantly different information is an important, yet understudied, open problem in exploratory data mining. Whether the goal is to select the most informative result for analysis, or to decide which mining approach will most likely provide the most novel insight, it is essential that we can tell how *different* the information is that *different* results by possibly *different* methods provide. In this paper we take a first step towards comparing exploratory data mining results on binary data. We propose to meaningfully convert results into sets of noisy tiles, and compare between these sets by maximum entropy modelling and Kullback–Leibler divergence, well-founded notions from Information Theory. We so construct a measure that is highly flexible, and allows us to naturally include background knowledge, such that differences in results can be measured from the perspective of what a user already knows. Furthermore, adding to its interpretability, it coincides with Jaccard dissimilarity when we only consider exact tiles. Our approach provides a means to study and tell differences between results of different exploratory data mining methods. As an application, we show that our measure can also be used to identify which parts of results best re-describe other results. Furthermore, we study its use for iterative data mining, where

---

Responsible editor: Dimitrios Gunopulos, Donato Malerba, Michalis Vazirgiannis.

---

The research described in this paper builds upon and extends the work appearing in ECML PKDD'11 as [Tatti and Vreeken \(2011\)](#).

---

N. Tatti (✉) · J. Vreeken  
Advanced Database Research and Modelling, Department of Mathematics and Computer Science,  
University of Antwerp, Antwerp, Belgium  
e-mail: nikolaj.tatti@ua.ac.be

J. Vreeken  
e-mail: jilles.vreeken@ua.ac.be

one iteratively wants to find that result that will provide maximal novel information. Experimental evaluation shows our measure gives meaningful results, correctly identifies methods that are similar in nature, automatically provides sound redescriptions of results, and is highly applicable for iterative data mining.

**Keywords** Exploratory data mining · Distance · Maximum entropy model · Iterative data mining · Tiling · Binary data

## 1 Introduction

Deciding whether the results of different mining algorithms provide significantly different information is an important, yet understudied, open problem in exploratory data mining. Whether we want to select the most promising result for analysis by an expert, or decide which mining approach we should apply next in order to most likely gain novel insight, we need to be able to tell how different the information is that is provided by different results, possibly from different methods. However, while the comparison of results is a well-studied topic in statistics, it has received much less attention in the knowledge discovery community.

Clearly, any dataset only contains a limited amount of knowledge—which is the most that we can hope to discover from it. To extract this information, we have at our disposal an ever growing number of data mining algorithms. However, when analysing data we have to keep in mind that most data mining results are complex, and their analysis and validation hence often requires considerable effort and/or cost. So, simply applying ‘all’ methods and having an expert analyse ‘all’ results is not a feasible approach to extract ‘all’ knowledge. Moreover, many of these results will be redundant, i.e. they will convey roughly the same information, and hence analysing ‘all’ obtained results will mostly require effort that will not provide extra insight.

Instead, we would ideally just select that result for analysis that will provide us the most new knowledge. In order to be able to do this, two basic requirements have to be met. First of all, we need to be able to measure how different two results are from an information-providing perspective; if they essentially provide the same information, we could just select one for processing. Second, we should be able to include our background knowledge, such that we can gauge the amount of information included in a result compared to what we already know.

Although an important practical problem, it has been surprisingly understudied in data mining. The main focus in exploratory data mining research has mostly been on developing techniques to discover structure, and not so much on how to compare between results of different methods. As a result there currently exist no general methods or theory to this end in the data mining literature.

For tasks where a formal objective is available, it makes sense to straightforwardly use that objective for comparing fairly between the results of different methods. In classification, for instance, we can use accuracy, or AUC scores—as well as that we can measure how differently classifiers behave, for instance in ensembles (Kuncheva and Whitaker 2003). In fact, in learning tasks in general, there typically are clearly defined goals, and hence different methods can be compared on their performance on that particular task.

For exploratory data mining, however, there is no formal common goal: the main goal is the *exploration* of the data at hand, and hence basically poses the rather general question ‘what can you tell me about my data?’. In fact, any type of data mining result that provides the analyst novel insight in the data, and the process that generated it, is potentially useful; regardless of whether the provided result of that method is, for example, a clustering, some subspace clusters, a collection of itemsets or correlated patterns, a decision tree, or even the identification of some statistical properties; as long as the analyst can obtain insight from that result, any data mining method can provide a valid (partial) answer to this question. Clearly, however, these methods do not share a common formal goal, and hence their results that are not easily comparable. The core of the problem is that comparing between methods is thus like comparing *apples to oranges*: a clustering is a different result than a set of itemsets, which are, in turn, different from a classifier, set of subgroups, etc. So, in order to make a sensible comparison, we need to find a common language.

In this regard, the comparison of complex objects, e.g. of datasets (Tatti 2007; Vreeken et al. 2007), is related. Our setting, however, is more general, as now we do not want to compare between one type of complex object, but want to consider a very rich class of objects—a class potentially consisting of *any* data mining result. Arguably, some of the most general complex objects to compare between are probability distributions. Statistics and information theory provide us tools for measuring differences between distributions, such as Kullback–Leibler divergence (Cover and Thomas 2006). Data mining results, however, rarely are probability distributions, and if they are, not necessarily for the same random variable; making these tools unsuited for direct application for our goal.

A simple yet important observation is that any mining result essentially identifies some properties of the dataset at hand. In fact, in an abstract way, we can identify all datasets for which these properties hold. This is an important notion, as it provides us with a way to compare between results of different methods: if two results provide the same information, they identify the same subspace of possible datasets, and the more different the information two results give, the smaller the overlap between the sets of possible datasets will be.

As such, a straightforward approach for measuring similarity between results would be to simply count the number of possible datasets per result, and regard the relative number in the overlap—the larger this number, the more alike the information provided. However, the most basic situation of rectangles full of ones aside, counting databases is far from trivial. Instead of counting, we can view our problem more generally, and regard a result as implicitly defining a probability distribution over datasets. And hence, if we can model these distributions, we can use standard tools from statistics to compare between data mining results fair and square.

In this paper, we propose to translate data mining results into probability distributions over datasets using the maximum entropy principle (Csiszár 1975). This allows us to uniquely identify that probabilistic model that makes optimal use of the information provided by a result, but is least biased otherwise. By subsequently measuring the Kullback–Leibler divergence between the obtained distributions, we can tell how different two results are from an information perspective. Furthermore, besides directly measuring the information content between data mining results, we can also

incorporate background knowledge into these models, and so score and compare informativeness of results from specific points of view.

Finding these probability distributions, and conditioning them using data mining results, however, is far from trivial in general. Here, we give a proof of concept of this approach for binary data, for which the basics of maximum entropy modelling are available (De Bie 2011b). We show that many types of results of exploratory data mining approaches on binary data can easily and meaningfully be translated into sets of noisy tiles: combinations of rows and columns, for which we know the density of 1s. We show we can efficiently acquire the maximum entropy distribution over datasets given such sets of noisy tiles, and construct a flexible measure that uses KL divergence by which we can compare between results.

To show that our measure works in practice, we compare between the results of ten different exploratory data mining methods, including (bi-)clusters, subspace clusters, sets of tiles, and sets of (frequent) itemsets. We further give a theoretic framework to mine for redescrptions. That is, given a (sub)set of noisy tiles from one result, we can identify the set of tiles from another result that best approximates the same information. Moreover, we show that our measure is highly effective for iterative data mining: given a (large) collection of results, we can efficiently discover which result provides the most novel information, and which hence is a likely candidate to be analysed next. Experiments show our measure works well in practice: dissimilarity converges to 0 when results approximate each other, methodologically close methods are correctly grouped together, sensible redescrptions for tile-sets are obtained, and meaningful orders of analysis are provided. In other words, we give an approach by which we can meaningfully *mix* apples and oranges, and compare between them fairly.

In this paper we build upon and extend the work published as (Tatti and Vreeken 2011). Here, we discuss the theory, properties, and choices of our measure in closer detail, as well as the limitations of this proof-of-concept measure. Most importantly, we investigate the practical application of our measure for both mining redescrptions of (partial) results, and for application to the end of iterative data mining—giving algorithms for both these settings, and providing experimental validation that they provide meaningful results.

The road map of this paper is as follows. Next, in Sect. 2 we give the notation and preliminaries we use throughout the paper, and discuss how to convert results on binary data into sets of tiles in Sect. 3. Section 4 details how we can build a global model from a set of tiles, which we use in Sect. 5 to define a measure to compare such sets. In Sect. 6 we subsequently use this measure for redescrbing sets of tiles, as well as to iteratively identify the most informative result. We discuss related work in Sect. 7, and we evaluate our measure empirically in Sect. 8. We round up with discussion and conclusions in Sects. 9 and 10. We give the proofs, such as for NP-completeness, in Appendix A.

## 2 Preliminaries

In this section, we define the preliminaries we will use in subsequent sections.

A *binary dataset*  $D$  is a binary matrix of size  $n$ -by- $m$  consisting of  $n$  rows, or transactions, and  $m$  columns, or attributes. A row is simply a binary vector of size  $m$ .

We assume that both the rows and columns of  $D$  have unique integer identifiers such that we can easily refer to them. We denote the  $(i, j)$ <sup>th</sup> entry of  $D$  by  $D(i, j)$ . We denote the space of all  $n$ -by- $m$  binary datasets by  $\mathcal{D}$ .

Given two distributions,  $p$  and  $q$ , defined over  $\mathcal{D}$ , we define entropy as

$$H(p) = - \sum_{D \in \mathcal{D}} p(D) \log p(D)$$

and Kullback–Leibler divergence as

$$KL(p \parallel q) = \sum_{D \in \mathcal{D}} p(D) \log \frac{p(D)}{q(D)}.$$

All logarithms are natural logarithms, to base  $e$ . We employ the usual convention of  $0 \log 0 = 0$ .

### 3 Results on binary data as noisy tiles

As discussed in Sect. 1, in this paper we give a proof of concept for measuring differences between data mining results by first converting these results into maximum entropy distributions over datasets, and subsequently measuring the difference in information content between these distributions. Slightly more formally, the first step entails inferring a maximum entropy distribution for the provided background knowledge. The background knowledge, in this case, being the data mining result at hand. To do so, we need theory on how to do this for the particular type of background knowledge—which is far from trivial in general, and hence there exists no theory for converting data mining results in general into (maximum entropy) probability distributions.

Recently, however, [De Bie \(2011b\)](#) formalised the theory for inferring the maximum entropy distribution for a binary dataset given so-called *noisy tiles* as background knowledge. We employ this result for our proof of concept, and hence focus on measuring differences between data mining results on binary data. While certainly not all data in the world is binary, surprisingly many (exploratory) data mining methods are aimed at extracting significant structure from 0 to 1 data.

First of all, there are approaches that identify whether data exhibits particular structure. As an example for binary data, the most simple example is density: what is the relative number of 1s of the data. A more intricate type of structure we can identify in this type of data is bandedness ([Garriga et al. 2011](#)): the property that the rows and columns of the data can be permuted such that the non-zero entries exhibit a staircase pattern of overlapping rows. Nestedness ([Mannila and Terzi 2007](#)), is the property that for every pair of rows of a 0–1 dataset one row is either a superset or subset of the other.

Perhaps the most well-known example for analysing binary data is frequent item-set mining ([Agrawal and Srikant 1994](#)). Given appropriate distance functions, binary data can be clustered ([MacQueen 1967](#)), or subspace clusters (clusters within specific subsets of dimensions) ([Aggarwal et al. 1999](#); [Müller et al. 2009](#)) can be identified.

Both traditional pattern mining and sub-space clustering typically provide incredibly many results. In response, a recent development is pattern *set* mining, where the goal is not to find all patterns (e.g. itemsets) that satisfy some conditions (e.g. frequency), but to instead find small, non-redundant, high-quality *sets* of patterns that together generalise the data well (Siebes et al. 2006; Miettinen et al. 2008; Knobbe and Ho 2006). An early proponent of this approach is Tiling (Geerts et al. 2004), which has the goal of finding large tiles, i.e. itemsets that cover many 1s in the data. Kontonasios and De Bie (2010) proposed to mine for *noisy* tiles that are surprising given a background model of the data. We discuss mining sets of patterns or tiles, as well as mining binary data in general, in more detail in Sect. 7.

Interestingly, as we will show below, many data mining results on binary data, including clustering and pattern mining results, can be translated into noisy tiles without much, or any loss of information. Before we discuss how different results can be translated, we have to formally introduce the concept of tiles.

A *tile*  $T = (t(T), a(T))$  is a tuple consisting of two lists. The first list,  $t(T)$ , is a set of integers between 1 and  $n$  representing the transaction ids of  $T$ . The second list,  $a(T)$ , is a set of integers between 1 and  $m$  representing the attribute ids. We define  $s(T)$  to give us all entries of  $T$ , that is, the Cartesian product of  $t(T)$  and  $a(T)$ ,  $s(T) = \{(i, j) \mid i \in t(T), j \in a(T)\}$ . The number of entries of  $s(T)$ ,  $|s(T)|$ , we refer to as the *area* of a tile  $T$ . Given a tile set  $\mathcal{T}$  we also define  $s(\mathcal{T}) = \bigcup_{T \in \mathcal{T}} s(T)$  to give us the list of all entries  $\mathcal{T}$  covers.

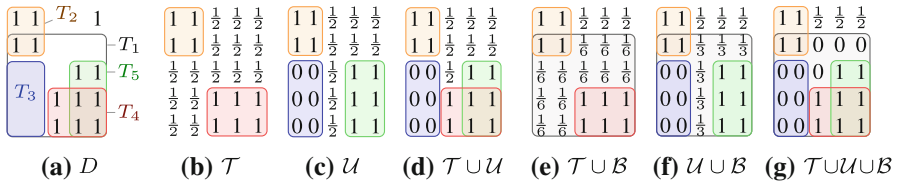
Given a tile  $T$  and a dataset  $D$  we define a frequency  $fr(T; D)$  to be the proportion of ones in  $D$  corresponding to the entries identified by  $T$ ,

$$fr(T; D) = \frac{1}{|s(T)|} \sum_{i \in t(T)} \sum_{j \in a(T)} D(i, j).$$

*Example 1* Assume that we are given a set of text documents, for example, a set of abstracts of accepted papers at a computer science conference. We can transform these documents into a binary dataset using a bag-of-words representation. That is, each item represents a particular word, and every document is represented by a binary vector  $t$  with entries  $t_i = 1$  if and only if the corresponding word  $i$  occurs in the abstract. Now, let  $T$  be a tile in this dataset. Then the transactions of  $T$  correspond to specific abstracts, while the columns, or the items of  $T$  represent specific words. The frequency of 1s in  $T$  is simply the relative number of these words occurring in these documents. For example, when we run the KRIMP algorithm on such a dataset (see Sect. 8 for more details on this algorithm), we see it finds (among others) tiles containing the words  $\{large, database\}$ ,  $\{algorithm, result, set, experiment\}$ , and  $\{synthetic, real\}$ . In this case, for each of the discovered tiles the corresponding frequencies are 1.0, and hence we know that each of the documents identified by these tiles contains all of the corresponding words.

Let  $p$  be a distribution defined over  $\mathcal{D}$ , the space of all  $n$ -by- $m$  datasets. We define the frequency of a tile to be the average frequency with respect to  $p$ ,

$$fr(T; p) = \sum_{D \in \mathcal{D}} p(D) fr(T; D).$$



**Fig. 1** Toy example of a dataset, and maximum entropy models for this data given different sets of tiles, as used in Examples 2–6. **a** Toy dataset  $D$  and five tiles,  $T_1, \dots, T_5$ , identified therein, **b** the maximum entropy model for  $D$  given tile set  $\mathcal{T} = \{T_2, T_4\}$ , **c** the maximum entropy model for  $D$  given tile set  $\mathcal{U} = \{T_2, T_3, T_5\}$ , **d** the maximum entropy model for  $D$  given  $\mathcal{T} \cup \mathcal{U}$ , **e** the maximum entropy model for  $D$  given  $\mathcal{T} \cup \mathcal{B}$ , with  $\mathcal{B} = \{T_1\}$ , **f** the maximum entropy model for  $D$  given tile set  $\mathcal{U} \cup \mathcal{B}$ , and **g** the maximum entropy model for  $D$  given tile set  $\mathcal{T} \cup \mathcal{U} \cup \mathcal{B}$ . The cell entries in (b–g) are given by Theorem 2

We can also express the frequency directly by this distribution.

**Lemma 1** *Given a distribution  $p$  and a tile  $T$ , the frequency is equal to*

$$fr(T; p) = \frac{1}{|s(T)|} \sum_{(i,j) \in s(T)} p((i, j) = 1),$$

where  $p((i, j) = 1)$  is the probability of a dataset having 1 as  $(i, j)$ th entry.

We say a tile is *exact* if its frequency is 0 or 1, and otherwise say it is *noisy*.

**Corollary 1** *For an exact tile  $T$ ,  $p((i, j) = 1) = fr(T; p)$ , where  $(i, j) \in s(T)$ .*

**Example 2** Consider a dataset  $D$  given in Fig. 1a. We consider five different tiles,

- $T_1 = (2, \dots, 5) \times (1, \dots, 5)$  of area 20 and covering 10 ones of  $D$ ,
- $T_2 = (1, 2) \times (1, 2)$  of area 4 and covering 4 ones,
- $T_3 = (3, 4, 5) \times (1, 2)$  of area 6 and covering 0 ones,
- $T_4 = (4, 5) \times (3, 4, 5)$  of area 6 and covering 6 ones,
- and  $T_5 = (3, 4, 5) \times (4, 5)$  of area 6 and also covering 6 ones.

Their subsequent frequencies are hence  $fr(T_1; D) = 10/20 = 1/2$  for  $T_1$ , while  $fr(T_2; D) = fr(T_4; D) = fr(T_5; D) = 1$ , and  $fr(T_3; D) = 0$ . By definition,  $T_1$  is a noisy tile, while  $T_2, \dots, T_5$  are exact tiles.

There are numerous techniques for mining tiles and sets from a database, but we observe that a large number of other statistics and mining results obtained on binary data can also be naturally described using sets of (noisy) tiles:

- *itemsets*: any itemset can be converted into a tile by taking the supporting transactions. For standard frequent itemsets this results in an exact tile, for fault-tolerant (or noisy) itemsets this results in a noisy tile. Thus, an itemset collection can be converted into a tile set.
- *clustering*: Given a clustering, we can construct a tile set in two different ways. The first way is to represent each cluster by a single tile, consisting of the transactions in the cluster and all the items. The other way is to transform each cluster into  $m$



tiles, where  $m$  is the number of items in the data. Each tile corresponds to an item and the transactions associated with the cluster. The frequency corresponding to a tile then simply is the relative occurrence of that item in that cluster, i.e. its mean value. This is particularly natural for  $k$ -means, since a centroid then corresponds to the column means of the corresponding transactions.

- *bi-clustering* and *subspace-clustering*: both subspace clusters (Aggarwal et al. 1999) and bi-clusters (Pensa et al. 2005) are sets of transactions and columns. Hence, we can naturally represent these results by equivalent tiles.
- *data density*: the density of the data is equal to the frequency of a tile containing the whole data.
- *column and row margins*: the margin of a column  $i$ , i.e. its frequency, can be expressed with a single (noisy) tile containing the column  $i$  and all rows  $r \in D$ . Analogously, we can express row margins by creating a tile per row over all columns.

As such, we can convert a wide range of results on binary data into sets of (noisy) tiles—and retain most, if not all, information provided by these results. Note, however, that this list is far from complete: we cannot convert every result on binary data into (sets of) (noisy) tiles. For instance, we can currently not model structures *within* a tile beyond its density, and hence in this proof-of-concept we disregard statistics that go beyond density, such as Lazarus counts (Fortelius et al. 2006), or nestedness (Mannila and Terzi 2007). We discuss converting data mining results into tiles, and the limitations of the current proof-of-concept, in more detail in Sect. 9.

#### 4 Building global models from tiles

To meet our goal, we have to construct a statistically sound technique for comparing two sets of tiles. In this section we construct a global model for datasets using the given tiles. We will use these models for comparing the tile sets.

Consider that we are given a tile set  $\mathcal{T}$ , and for each tile  $T \in \mathcal{T}$  we are also given a frequency  $\alpha_T$ . Typically, the frequencies are obtained from the data at hand,  $\alpha_T = fr(T; D_{in})$ , but this is not a necessary condition. The tiles convey local information about the data  $D_{in}$  and our goal is to infer a distribution  $p$  over  $\mathcal{D}$ , that is, how probable data set  $D \in \mathcal{D}$  is given a tile set  $\mathcal{T}$ . If the information at hand defines the data set uniquely, then  $p(D) = 1$  if and only if  $D = D_{in}$ .

To derive the model, we use a well-founded notion from information theory, the maximum entropy principle (Csiszár 1975; Jaynes 1982). Roughly speaking, by maximum entropy, we incorporate the given information into a distribution, yet further making it as evenly spread as possible. To define the distribution, we first define the space of distribution candidates. That is, the space of those distributions that produce the same frequencies for the given tiles,  $\mathcal{P} = \{p \mid fr(T; p) = \alpha_T, \text{ for all } T \in \mathcal{T}\}$ . In other words,  $\mathcal{P}$  contains all distributions that explain the frequencies  $\alpha_T$ . From this set, we select one distribution, which we denote by  $p_{\mathcal{T}}^*$ , such that  $p_{\mathcal{T}}^*$  maximises the entropy,  $H(p_{\mathcal{T}}^*) \geq H(p)$  for any  $p \in \mathcal{P}$ .

We will abuse notation and write  $H(\mathcal{T})$  where we mean  $H(p_{\mathcal{T}}^*)$ . Similarly we write  $KL(\mathcal{T} \parallel \mathcal{U})$  to mean  $KL(p_{\mathcal{T}}^* \parallel p_{\mathcal{U}}^*)$ , where both  $\mathcal{T}$  and  $\mathcal{U}$  are tile sets.



A classic theorem states that the maximum entropy distribution  $p^*$  can be written in an exponential form—which ensures generality, and as we show below, provides us with a practical way for inferring the model.

**Theorem 1 (Theorem 3.1 in (Csiszár 1975))** *Given a tile set  $\mathcal{T}$ , a distribution  $p^*$  is the maximum entropy distribution if and only if it can be written as*

$$p^*(D) \propto \begin{cases} \exp(\sum_{T \in \mathcal{T}} \lambda_T |s(T)| fr(T; D)) & D \notin \mathcal{Z} \\ 0 & D \in \mathcal{Z}, \end{cases}$$

where  $\lambda_T$  is a certain weight for  $fr(T; D)$  and  $\mathcal{Z}$  is a collection of datasets such that  $p(D) = 0$  for each  $p \in \mathcal{P}$ .

The next theorem allows to factorize the distribution  $p^*$  into a product of Bernoulli random variables, each variable representing a single entry in the dataset. Such a representation gives us a practical way for inferring the model.

**Theorem 2** *Let  $\mathcal{T}$  be a tile set. Write  $\mathcal{T}(i, j) = \{T \in \mathcal{T} \mid (i, j) \in s(T)\}$  to be the subset of  $\mathcal{T}$  containing the tiles that cover an entry  $(i, j)$ . Then, the maximum entropy distribution can be factorised as  $p^*(D) = \prod_{i,j} p^*((i, j) = D(i, j))$ , where*

$$p^*((i, j) = 1) = \frac{\exp(\sum_{T \in \mathcal{T}(i,j)} \lambda_T)}{\exp(\sum_{T \in \mathcal{T}(i,j)} \lambda_T) + 1} \quad \text{or} \quad p^*((i, j) = 0) = 1.$$

*Proof* See Appendix A. □

Theorem 2 allows to represent  $p^*$  as Bernoulli variables. We should stress that this is a different model than assuming independence between items in a random transaction. Our next step is to discover the correct frequencies for these variables. Here we use a variant of a well-known Iterative Scaling algorithm (Darroch and Ratcliff 1972). The algorithm is given in Algorithm 1. Informally said, given a tile  $T$ , the algorithm updates the probabilities in the model such that the expected frequency of 1s within the area defined by  $T$  becomes equal to  $\alpha_T$ . We do this for each tile. However, updating the model to satisfy a single tile might change the frequency of another tile, and hence we need several passes. The update phase is done by modifying the weight  $\lambda_T$  (given in Theorem 1). For the proof of convergence see Theorem 3.2 in (Csiszár 1975).

Instead of representing a distribution with the exponential form given in Theorem 1 we prefer working with the matrix form given in Theorem 2. We need to show that the update phase in Algorithm 1 actually corresponds modifying the weight  $\lambda_T$ . In order to do that let  $p$  be a distribution having the exponential form, let  $T$  be a tile let  $q$  be a distribution  $q(D) \propto \exp(\lambda fr(T; D)) p(D)$ , that is  $q$  is the distribution for which  $\lambda_T$  is shifted by  $\lambda$ . Let us define

$$u(y, x) = \frac{xy}{1 - y(1 - x)}. \tag{1}$$

---

**Algorithm 1:** Iterative scaling for solving the MaxEnt distribution

---

**input** : tile set  $\mathcal{T}$ , target frequencies  $\{\alpha_T\}$   
**output** : Maximum entropy distribution  $p$

- 1  $p \leftarrow$  a  $n$ -by- $m$  matrix with values  $1/2$ ;
- 2 **foreach**  $T \in \mathcal{T}, \alpha_T = 0, 1$  **do**  $p(i, j) \leftarrow \alpha_T$  for all  $(i, j) \in s(T)$ ;
- 3 **while** not converged **do**
- 4     **foreach**  $T \in \mathcal{T}, 0 < \alpha_T < 1$  **do**
- 5          $f \leftarrow \hat{f}_r(T; p)$ ;
- 6         find  $x$  such that  $f = \sum_{(i,j) \in s(T)} u(p(i, j), x)$ ; { see Equation 1 }
- 7          $p(i, j) \leftarrow u(p(i, j), x)$  for all  $(i, j) \in s(T)$ ;

---

For the sake of clarity, let us write  $a = p((i, j) = 1)$  and  $b = q((i, j) = 1)$ . We claim that  $b = u(a, e^\lambda)$  if  $(i, j) \in s(T)$  and  $a = b$  otherwise. The latter claim follows directly from Theorem 2 since if  $(i, j) \notin s(T)$ , then the fraction in Theorem 2 does not contain  $\lambda_T$ . Assume that  $(i, j) \in s(T)$ . Then

$$b = \frac{e^\lambda c}{e^\lambda c + 1} \quad \text{and} \quad a = \frac{c}{c + 1}, \quad \text{where} \quad c = \sum_{T \in \mathcal{T}(i,j)} \lambda_T.$$

Expressing  $b$  using  $a$  implies that  $b = u(a, e^\lambda)$ . In order to find the correct  $e^\lambda$  we use interpolation search using a linear interpolation.

It turns out, that if the given tiles are exact, the maximum entropy distribution has a very simple form. The Bernoulli variable corresponding to the  $(i, j)$ th entry is always 1 (or 0), if that entry is covered by an exact tile, i.e. with frequency 1 (or 0). Otherwise, the variable is equal to a fair coin toss. This form will allow us to express distances between sets of exact tiles in the next section.

**Theorem 3** *Let  $\mathcal{T}$  be a collection of exact tiles and let  $\alpha_T$  be the desired frequency of a tile  $T \in \mathcal{T}$ . Then*

$$p_{\mathcal{T}}^*((i, j) = 1) = \begin{cases} \alpha_T & \text{if there exists } T \in \mathcal{T} \text{ such that } (i, j) \in s(T) \\ 1/2 & \text{otherwise.} \end{cases}$$

*Proof* See Appendix A. □

In essence, this theorem states that in the maximum entropy model for a dataset given set  $\mathcal{T}$  of exact tiles, the probability  $p^*$  of observing a 1 in a specified cell  $(i, j)$  is either 0, or 1 if this cell maps to one of the tiles in  $\mathcal{T}$ —and otherwise, the probability is  $1/2$  as we cannot infer anything about this cell from  $\mathcal{T}$ .

*Example 3* Let us continue Example 2. Consider the following three tile sets  $\mathcal{T} = \{T_2, T_4\}$ ,  $\mathcal{U} = \{T_2, T_3, T_5\}$ , and  $\mathcal{B} = \{T_1\}$ . The corresponding maximum entropy models are given in Fig. 1, such that each entry represents the probability  $p^*((i, j) = 1)$ . As the sets  $\mathcal{T}$  and  $\mathcal{U}$  contain only exact tiles, Theorem 3 implies that the entries for the models  $p_{\mathcal{T}}^*$ ,  $p_{\mathcal{U}}^*$ , and  $p_{\mathcal{T} \cup \mathcal{U}}^*$  are either 0, 1, or  $1/2$ .

Consider  $p_{\mathcal{T} \cup \mathcal{B}}^*$ . Corollary 1 states that entries in  $s(T_2)$  and  $s(T_4)$  should be 1, since both tiles are exact. From  $T_1$ , we know that there are 10 ones, yet  $T_2$  and  $T_4$  account only for 8. Hence, there should be 2 ones in the 12 entries outside of  $\mathcal{T}$ , on average. We aim to be as fair as possible, hence we spread uniformly, giving us probabilities  $2/12 = 1/6$  for the unaccounted entries in  $T_1$ . For  $p_{\mathcal{U} \cup \mathcal{B}}^*$ , there are 2 unaccounted 1s in 6 entries, giving us  $2/6 = 1/3$ . Finally, all 1s are accounted for in  $p_{\mathcal{T} \cup \mathcal{U} \cup \mathcal{B}}^*$ . Outside of  $\mathcal{T} \cup \mathcal{U} \cup \mathcal{B}$  we have no information, hence these probabilities default to  $1/2$ .

### 5 Comparing sets of tiles

Now that we have a technique for incorporating the information contained within a set of tiles into a probabilistic model, we can subsequently use these models to compare between the sets of tiles.

We assume that we are given three tile sets  $\mathcal{T}$ ,  $\mathcal{U}$ , and  $\mathcal{B}$ . Let tile set  $\mathcal{B}$  contain the background information. Typically, this information would be simple, like column margins, row margins, or just the proportions of ones in the whole dataset.  $\mathcal{B}$  can be also be empty, if we do not have or wish to use any background knowledge. Our goal is now to compute the distance between  $\mathcal{T}$  and  $\mathcal{U}$  given  $\mathcal{B}$ . We assume that the frequencies for the tile sets we are given are mutually consistent; which is automatically guaranteed if the frequencies for all three tile sets are computed from a single dataset. Now, let  $\mathcal{M} = \mathcal{T} \cup \mathcal{U} \cup \mathcal{B}$  be the collection containing all tiles. We define the distance between  $\mathcal{T}$  and  $\mathcal{U}$ , w.r.t.  $\mathcal{B}$ , as

$$d(\mathcal{T}, \mathcal{U}; \mathcal{B}) = \frac{KL(\mathcal{M} \parallel \mathcal{U} \cup \mathcal{B}) + KL(\mathcal{M} \parallel \mathcal{T} \cup \mathcal{B})}{KL(\mathcal{M} \parallel \mathcal{B})}.$$

If  $KL(\mathcal{M} \parallel \mathcal{B}) = 0$ , we define  $d(\mathcal{T}, \mathcal{U}; \mathcal{B}) = 1$ . A direct application of Theorem 2 leads to the following theorem which in turns imply that we can compute the distance in  $O(nm)$  time.

**Theorem 4** *Let  $\mathcal{A}$  and  $\mathcal{B}$  be two sets of tiles such that  $\mathcal{B} \subseteq \mathcal{A}$ . Then*

$$KL(\mathcal{A} \parallel \mathcal{B}) = \sum_{i,j} \sum_{v=0,1} p_{\mathcal{A}}^*((i, j) = v) \log \frac{p_{\mathcal{A}}^*((i, j) = v)}{p_{\mathcal{B}}^*((i, j) = v)}.$$

If the given tiles are exact, the distance has a simple interpretable form; namely, the distance can be expressed with Jaccard similarity.

**Theorem 5** *Assume three tile collections  $\mathcal{T}$ ,  $\mathcal{U}$ , and  $\mathcal{B}$  with exact frequencies  $\{\alpha_{\mathcal{T}}\}$ ,  $\{\beta_{\mathcal{U}}\}$ , and  $\{\gamma_{\mathcal{B}}\}$ . Define  $X = s(\mathcal{T}) \setminus s(\mathcal{B})$  and  $Y = s(\mathcal{U}) \setminus s(\mathcal{B})$ . Then  $d(\mathcal{T}, \mathcal{U}; \mathcal{B}) = 1 - |X \cap Y|/|X \cup Y|$  for  $|X \cup Y| > 0$  and  $d(\mathcal{T}, \mathcal{U}; \mathcal{B}) = 1$  for  $|X \cup Y| = 0$ .*

*Proof* See Appendix A. □

In other words, the theorem states that if we only consider sets of exact tiles, we can easily compute (and interpret!) the dissimilarity score  $d$  between them, as our measure coincides with Jaccard dissimilarity.

*Example 4* Let us continue Example 3. To compute  $d(\mathcal{T}, \mathcal{U}; \emptyset)$  we first note that  $\mathcal{T}$  and  $\mathcal{U}$  only have exact tiles, and hence we can use Theorem 5. So, we have  $|s(\mathcal{T}) \setminus s(\mathcal{U})| = 2$ ,  $|s(\mathcal{U}) \setminus s(\mathcal{T})| = 8$ , and  $|s(\mathcal{T}) \cup s(\mathcal{U})| = 18$ . And hence, the distance  $d(\mathcal{T}, \mathcal{U}; \emptyset) = (2 + 8)/18 = 5/9$ .

Next, let us use  $\mathcal{M}$  as a shorthand for  $\mathcal{T} \cup \mathcal{U} \cup \mathcal{B}$ , i.e.  $\mathcal{M} = \mathcal{T} \cup \mathcal{U} \cup \mathcal{B}$ . To compute  $d(\mathcal{T}, \mathcal{U}; \mathcal{B})$ , note that

$$KL(\mathcal{M} \parallel \mathcal{T} \cup \mathcal{B}) = 2 \log(6) + 10 \log(6/5) \approx 5.4067.$$

where the first term represents the positive entries in  $\mathcal{M}$  and the second term the negative entries in  $\mathcal{M}$ . Similarly,  $KL(\mathcal{M} \parallel \mathcal{B}) \approx 15.2$ , and  $KL(\mathcal{M} \parallel \mathcal{U} \cup \mathcal{B}) \approx 3.8$ . Consequently, the distance between  $\mathcal{T}$  and  $\mathcal{U}$  in light of  $\mathcal{B}$  is

$$d(\mathcal{T}, \mathcal{U}; \mathcal{B}) = \frac{3.8 + 5.4}{15.2} \approx 0.6$$

which is slightly larger than  $d(\mathcal{T}, \mathcal{U}; \emptyset) \approx 0.56$ . This is due to the fact that adding  $\mathcal{B}$ , which effectively states the data is sparse, to  $\mathcal{T}$  makes the probability of encountering a 1 at (3, 4) and (3, 5) in the model less likely. Hence, given that background knowledge, and regarding  $\mathcal{U}$ , we are more surprised to find that the values of these entries are indeed ones.

Next, we discuss some further properties of our measure. We start by showing the measurement is upper bounded by 2 for noisy tiles, and 1 if we only consider exact tiles.

**Theorem 6** *Assume three tile collections  $\mathcal{T}$ ,  $\mathcal{U}$ , and  $\mathcal{B}$ . Then  $d(\mathcal{T}, \mathcal{U}; \mathcal{B}) \leq 2$ . If  $\mathcal{T}$ ,  $\mathcal{U}$ , and  $\mathcal{B}$  consist only of exact tiles, then  $d(\mathcal{T}, \mathcal{U}; \mathcal{B}) \leq 1$ .*

*Proof* See Appendix A. □

Before, in Theorem 5 we showed that when all tiles are exact and  $s(\mathcal{T}) \cap s(\mathcal{U}) \subseteq s(\mathcal{B})$ , that is,  $\mathcal{T}$  does not provide any new information about  $\mathcal{U}$ , then the distance is equal to 1. We can extend this property to noisy tiles.

**Theorem 7** *Assume three tile collections  $\mathcal{T}$ ,  $\mathcal{U}$ , and  $\mathcal{B}$ . Assume that  $\mathcal{B}$  consists of exact tiles and  $s(\mathcal{T}) \cap s(\mathcal{U}) \subseteq s(\mathcal{B})$ . Then  $d(\mathcal{T}, \mathcal{U}; \mathcal{B}) = 1$ .*

*Proof* See Appendix A. □

Exploiting the fact that our measure coincides with Jaccard dissimilarity when considering exact tiles, our measure upholds the triangle inequality when all tile sets consist of exact tiles.

**Theorem 8** *Assume four tile collections  $\mathcal{S}$ ,  $\mathcal{T}$ ,  $\mathcal{U}$ , and  $\mathcal{B}$  consisting only of exact tiles. Then the distance satisfies the triangle inequality,*

$$d(\mathcal{T}, \mathcal{U}; \mathcal{B}) \leq d(\mathcal{T}, \mathcal{S}; \mathcal{B}) + d(\mathcal{S}, \mathcal{U}; \mathcal{B}).$$

*Proof* See Appendix A. □

Consequently, the distance is (almost) a metric when working only with exact tiles. The technical subtlety is that the distance can be 0 even if the tile sets are different. This will happen, if  $(s(\mathcal{T}) \cup s(\mathcal{U})) \setminus (s(\mathcal{T}) \cap s(\mathcal{U})) \subseteq s(\mathcal{B})$ . This highlights the fact that we are measuring the information hidden in the tile sets, not the actual tiles.

Next, we show that if we add an element from one tile set to the other, the distance we measure between them decreases.

**Theorem 9** *Assume three tile collections  $\mathcal{T}$ ,  $\mathcal{U}$ , and  $\mathcal{B}$ . Let  $U \in \mathcal{U}$ . Then*

$$d(\mathcal{T} \cup \{U\}, \mathcal{U}; \mathcal{B}) \leq d(\mathcal{T}, \mathcal{U}; \mathcal{B}).$$

*Proof* See Appendix A. □

In other words, if we add tiles from the right-hand tile set  $\mathcal{U}$  to the left-hand tile set  $\mathcal{T}$ , regardless of the background information  $\mathcal{B}$ , we decrease the dissimilarity  $d(\mathcal{T}, \mathcal{U}; \mathcal{B})$ , as  $\mathcal{T}$  contains more information about  $\mathcal{U}$ .

*Example 5* Consider the tile sets  $\mathcal{U}$  and  $\mathcal{M} = \mathcal{T} \cup \mathcal{U} \cup \mathcal{B}$  given in Fig. 1. The distance between  $\mathcal{U}$  and  $\mathcal{M}$  is  $6/22$ . The distance between  $\mathcal{U} \cup \{T_4\} = \mathcal{U} \cup \mathcal{T}$  and  $\mathcal{M}$  is  $3/22$ . Since  $T_4 \in \mathcal{M}$ , Theorem 10 states that the distance  $d(\mathcal{U}, \mathcal{M}; \emptyset)$  should be as large as  $d(\mathcal{U} \cup \{T_4\}, \mathcal{M}; \emptyset)$ , which is the case.

If  $\mathcal{U} \subseteq \mathcal{T}$ , then we can show that the smaller the distance, the larger the Kullback–Leibler is between  $\mathcal{U}$  and the background knowledge. We will use this property for iterative data mining in Sect. 6.2.

**Theorem 10** *Assume three tile collections  $\mathcal{T}$ ,  $\mathcal{U}$ , and  $\mathcal{B}$ . Let  $\mathcal{M} = \mathcal{T} \cup \mathcal{U} \cup \mathcal{B}$ . Assume that  $\mathcal{U} \cup \mathcal{B} \subseteq \mathcal{T} \cup \mathcal{B}$ . Then*

$$d(\mathcal{T}, \mathcal{U}; \mathcal{B}) = \frac{KL(\mathcal{M} \parallel \mathcal{U} \cup \mathcal{B})}{KL(\mathcal{M} \parallel \mathcal{B})} = 1 - \frac{KL(\mathcal{U} \cup \mathcal{B} \parallel \mathcal{B})}{KL(\mathcal{M} \parallel \mathcal{B})}.$$

*Proof* See Appendix A. □

Let us next consider an example of this property.

*Example 6* Consider the tile sets  $\mathcal{T}$  and  $\mathcal{U}$  given in Fig. 1. Let  $\mathcal{M} = \mathcal{T} \cup \mathcal{U}$ . We have  $KL(\mathcal{M} \parallel \mathcal{U}) = 2$ ,  $KL(\mathcal{U} \parallel \emptyset) = 16$ ,  $KL(\mathcal{M} \parallel \emptyset) = 18$ . The distance is then

$$d(\mathcal{U}, \mathcal{U} \cup \mathcal{T}; \emptyset) = \frac{KL(\mathcal{M} \parallel \mathcal{U})}{KL(\mathcal{M} \parallel \emptyset)} = \frac{2}{18} = 1 - \frac{16}{18} = 1 - \frac{KL(\mathcal{U} \parallel \emptyset)}{KL(\mathcal{M} \parallel \emptyset)}.$$

## 6 Applying the distance

Above, we were only concerned in finding out how much information two tile sets share. With our measure, we can quantify the difference in information provided by different tile sets, and hence, measure differences between results—a very useful application on itself. In this section we consider two more elaborate problems to which our measure can be applied.

## 6.1 Redescribing results by redescribing sets of tiles

The first application we consider is redescription of results. Opposed to traditional redescription mining (Ramakrishnan et al. 2004), where given a pattern  $X$  we want to find a syntactically different pattern  $Y$  that identifies the same rows in the database as  $X$ , we here study how well we can describe the information provided by one result using (parts of) another result. Between these two approaches, the underlying question is the same: are there different representations of the same information. Whereas in standard redescription mining the row-set of a pattern is regarded as ‘information’, we consider the detail given on the data (distribution) as information.

More formally, given two tile sets, say  $\mathcal{T}$  and  $\mathcal{U}$ , we want to find which tiles from  $\mathcal{U}$  best describe the information provided by  $\mathcal{T}$ . To this end, we can apply our distance measure as follows.

**Problem 1 (REDESCRIBE)** Given three sets of tiles  $\mathcal{T}$ ,  $\mathcal{U}$ , and  $\mathcal{B}$  with consistent frequencies, find a subset  $\mathcal{V} \subseteq \mathcal{U}$  such that  $d(\mathcal{V}, \mathcal{T}; \mathcal{B})$  is minimised.

As with many optimisation problems in data mining, it turns out that finding the best tile subset is computationally intractable.

**Theorem 11** *The decision version of REDESCRIBE is an NP-hard problem.*

*Proof* See Appendix A. □

Hence, we resort to a simple greedy heuristic to find good approximations for the redescription problem: we iteratively choose to add that tile to our selection that maximally reduces the difference to the target tile set. We give the pseudo-code for our redescription mining approach as Algorithm 2. As the number of tiles we can select is finite, so is the procedure. Moreover, we stop selecting new tiles when we cannot decrease the distance by adding more tiles.

---

**Algorithm 2:** FRUITS, finds redescription used in a tile set.

---

**input** : target tile set  $\mathcal{T}$ , candidate tile set  $\mathcal{S}$ , background knowledge  $\mathcal{B}$   
**output** :  $\mathcal{R}$  a subset of  $\mathcal{S}$  redescribing  $\mathcal{T}$

```

1  $\mathcal{R} \leftarrow \emptyset$ ;
2 while changes do
3    $d \leftarrow d(\mathcal{R}, \mathcal{T}; \mathcal{B})$ ;
4    $C \leftarrow \emptyset$ ;
5   foreach  $S \in \mathcal{S}$  do
6     if  $d(\mathcal{R} \cup \{C\}, \mathcal{T}; \mathcal{B}) \leq d$  then
7        $C \leftarrow S$ ;
8        $d \leftarrow d(\mathcal{R} \cup \{C\}, \mathcal{T}; \mathcal{B})$ ;
9   if  $C \neq \emptyset$  then add  $C$  to  $\mathcal{R}$ ; remove  $C$  from  $\mathcal{S}$ ;
10 return  $\mathcal{R}$ ;
```

---

Note that Algorithm 2 selects tiles in a particular order: we iteratively find the tile that minimises the distance to the information in the target tile set. As such, if needed,

one can only inspect the top- $k$  selected tiles of a redescription to see how result  $\mathcal{S}$  captures the most important information in  $\mathcal{T}$ .

Furthermore, by iteratively minimising the distance, if the candidate tile set  $\mathcal{S}$  contains elements of the target tile set  $\mathcal{T}$ , these tiles are likely to be selected—as those parts of the target information can be ‘redescribed’ exactly. Also note that by minimising the distance between  $\mathcal{R}$  and  $\mathcal{T}$  we will not simply select all tiles in  $\mathcal{S}$  that overlap with tiles in  $\mathcal{T}$ —even though by overlapping these tiles will provide some similar information, their non-overlapping parts will provide different information, and hence increase the distance between the two tile sets. As such, we automatically prevent the selection of tiles in  $\mathcal{S}$  that are much more specific (e.g. contain more items) than what  $\mathcal{T}$  describes.

## 6.2 Iterative data mining

Next, we apply our distance measure for application in an iterative data mining setting. The key idea of iterative data mining is that what one finds interesting depends on what one knows. As such, practitioners study results one at the time (Hanhijärvi et al. 2009; Mampaey et al. 2012). Once we have studied a result, we are no longer interested in other results that provide (approximately) the same information, and we rather want to be served the most interesting pattern with regard to what we now know.

As such, the key difference between iterative data mining and traditional pattern ranking is that we take into account what a user has already seen when determining what might be interesting. Consequently, if a result has a peculiar statistic, yet it can be (approximately) explained by the previously inspected results, we consider this result to be redundant and choose not present it to the user. To be more specific, let us assume that we ran  $k$  different algorithms on a single dataset  $D$ , and so obtained  $\mathcal{T}_1, \dots, \mathcal{T}_k$  different tile sets. Now, if we merge all tiles into one tile set  $\mathcal{T} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_k$ , this set represents all information we have assembled about  $D$ . Note however that  $\mathcal{T}$  is unordered, and will likely contain redundant tiles. We order the tiles by applying FITAMIN, which orders the tiles in a given tile set  $\mathcal{T}$  such that every tile provides maximal novel information with regard to the tiles ranked above it.

---

### Algorithm 3: FITAMIN, Find iteratively those tiles that are most informative

---

**input** : tile set  $\mathcal{T}$ , background knowledge  $\mathcal{B}$   
**output** : ordered list of tiles occurring in  $\mathcal{T}$

- 1  $\mathcal{L} \leftarrow \emptyset$ ;
- 2  $\mathcal{U} \leftarrow \mathcal{T}$ ;
- 3 **while**  $\mathcal{U} \neq \emptyset$  **do**
- 4      $T \leftarrow \arg \min_U \{d(\mathcal{L} \cup \{U\}, \mathcal{T}; \mathcal{B}) \mid U \in \mathcal{U}\}$ ;
- 5     add  $T$  to the end of  $\mathcal{L}$ ;
- 6     delete  $T$  from  $\mathcal{U}$ ;
- 7 **return**  $\mathcal{L}$ ;

---

The FITAMIN algorithm, for which we present the pseudo-code as Algorithm 3, starts with an empty list  $\mathcal{L}$  and selects a tile  $T \in \mathcal{T}$  which minimises the distance  $d(\mathcal{L} \cup \{U\}, \mathcal{T}; \mathcal{B})$ . We add  $T$  into  $\mathcal{L}$  and repeat the process until all tiles are ordered.



In order to understand the connection between this approach and iterative data mining, let us consider Theorem 10 which implies that the tile that minimises the distance, also maximises  $KL(\mathcal{L} \cup \mathcal{B} \cup \{T\} \parallel \mathcal{B})$ . Corollary 2 (given in Appendix) allows us to rewrite the divergence as a difference between two entropies

$$KL(\mathcal{L} \cup \mathcal{B} \cup \{T\} \parallel \mathcal{B}) = H(\mathcal{B}) - H(\mathcal{L} \cup \mathcal{B} \cup \{T\}).$$

Note that  $H(\mathcal{B})$  does not depend on  $T$  and so if we replace it with  $H(\mathcal{L} \cup \mathcal{B})$ , then we conclude that  $T$  must maximise

$$H(\mathcal{B} \cup \mathcal{L}) - H(\mathcal{L} \cup \mathcal{B} \cup \{T\}) = KL(\mathcal{L} \cup \mathcal{B} \cup \{T\} \parallel \mathcal{B} \cup \mathcal{L}),$$

where the equality follows from Corollary 2.

The divergence  $KL(\mathcal{L} \cup \mathcal{B} \cup \{T\} \parallel \mathcal{B} \cup \mathcal{L}) = 0$  if and only if the frequency of  $T$  obtained from the dataset is equal to the frequency obtained from the maximum entropy model constructed from the known tiles  $\mathcal{B} \cup \mathcal{L}$ . The divergence increases as these two frequencies become more different. Consequently, a tile  $T$  is the one that is most surprising given the known tiles.

## 7 Related work

To our knowledge, defining a distance between two *general* tile sets is a novel idea. However, there exist several techniques for comparing between datasets by comparing how the supports of patterns change between the datasets. Such proposals include a Mahalanobis distance between itemset collections (Tatti 2007) and a compression-based distance between itemsets (Vreeken et al. 2007). In addition, Hollmén et al. (2003) suggested using  $L_1$  distance between frequent itemset collections, where the missing frequencies were estimated by a support threshold.

From technical point of view, comparing pattern sets given background knowledge is akin to defining an interestingness measure based on deviation from the background knowledge. In fact, our approach for building a global maximum entropy model from tiles was inspired by the work of De Bie (2011b), where he builds a similar maximum entropy model from row and column margins (i.e. a Rasch model (Rasch 1960)) and uses it as a static null hypothesis to rank tiles. Further related proposals include iterative mining of patterns by empirical  $p$  values and randomisation (Hanhijärvi et al. 2009), and maximum entropy models based on itemsets (Wang and Parthasarathy 2006; Mampaey et al. 2012).

Several techniques have been proposed for mining sets of tiles. Geerts et al. (2004) suggested discovering tilings that cover as many ones as possible. Xiang et al. (2011) gave a method to mine (possibly noisy) tiles that cover ones while minimising a cost: the number of transactions and items needed to describe tiles. These methods focus on covering the ones in the data, alternatively, we can assess the quality of a tiling by statistical means. Gionis et al. (2004) suggested discovering hierarchical tiles by building a statistical model and optimising an MDL score. De Bie (2011b) gave a maximum entropy model based on column/row margins to rank tiles. Vreeken et al.

(2011) propose that the best set of tiles (or itemsets) is the tile set that compresses the dataset best.

An alternative approach for discovering tiles is to consider a Boolean matrix factorisation (Miettinen et al. 2008). That is, factorise the dataset into two low rank Boolean matrices, where the row vectors of the one matrix correspond to itemsets, while the column vectors of the other matrix correspond to *tid*-lists. The Boolean product of these matrices naturally defines a set of noisy tiles.

Compared to tiles, computing maximum entropy models based on itemsets is much more difficult. The reason for this is that there is no equivalent version of Lemma 1 for itemsets. In fact, computing an expected value of an itemset from a maximum entropy model is **PP**-hard (Tatti 2006). To avoid these problems, we can convert itemsets to exact tiles by considering their supporting transactions.

Redescribing tile sets is closely related to redescription mining, in which the idea is to find pairs of syntactically different patterns covering roughly the same transactions. Ramakrishnan et al. (2004) originally approached the problem by building decision trees. Other approaches include Boolean Formulae with no overlap (Gallo et al. 2008), and exact minimal redescrptions (Zaki and Ramakrishnan 2005). From a computational point of view, the difference between our problem and existing work is that whereas redescription mining aims to construct alternative patterns given a *single* target pattern, we on the other hand consider *sets* of target and candidate patterns, and aim to find the *subset* of patterns from candidates that together describe the target best.

Iterative data mining, in which one iteratively finds the most interesting result given what one already knows, and subsequently dynamically updates the background model, is a relatively new approach to data mining (De Bie 2011a). Hanhijärvi et al. (2009) propose to use swap-randomisation for acquiring empirical *p* values for patterns, keeping the margins of already selected patterns (approximately) fixed. By using an analytical model of the data, instead of having to sample many randomised datasets per pattern, we gain much computational efficiency. Whereas we model complete databases, Mampaey et al. (2012) formalise a maximum entropy model for rows of the data by iteratively choosing that itemset for which the frequency prediction is most off. Both models have advantages: whereas in our model it is not obvious how itemset frequencies can be easily incorporated, i.e. without specifying in which rows of the dataset these occur, in the row-model approach co-occurrences of patterns are difficult to detect, as we do not know *where* patterns occur.

## 8 Experiments

In this section we empirically evaluate our measure, applying it for measuring distances between results, redescrptions of results, and for iterative data mining.

### 8.1 Set up

We evaluate our measure on four publicly available real world datasets. The *Abstracts* dataset contains the abstracts of the papers accepted at ICDM up to 2007, where words have been stemmed and stop words removed (De Bie 2011b). The *DNA*

**Table 1** Number of tiles extracted by each of the considered methods

Dataset	$n$	$m$	Number of tiles per method									
			<i>clus</i>	<i>bicl</i>	<i>atcl</i>	<i>sscl</i>	<i>asso</i>	<i>tiling</i>	<i>hyper</i>	<i>itt</i>	<i>krimp</i>	<i>mtv</i>
Abstracts	859	3,933	5m	25	753	100	100	38	100	100	100	25
DNA	4,590	392	5m	25	56	100	100	32	100	100	100	100
Mammals	2,183	124	5m	25	28	100	91	3	100	2	100	14
Paleo	501	139	5m	25	514	100	100	100	100	71	85	14

amplification data is data on DNA copy number amplifications. Such copies activate oncogenes and are hallmarks of nearly all advanced tumours (Myllykangas et al. 2006). Amplified genes represent attractive targets for therapy, diagnostics and prognostics. The *Mammals* presence data consists of presence records of European mammals<sup>1</sup> within geographical areas of  $50 \times 50$  km (Mitchell-Jones et al. 1999). Finally, *Paleo* contains information on fossil records<sup>2</sup> found at specific palaeontological sites in Europe (Fortelius et al. 2006).

We provide our prototype implementation for research purposes<sup>3</sup>. Computing a single distance typically takes a few seconds—up to maximally two minutes for the *Abstracts* dataset, when considering the most complex sets of tiles, and most detailed background knowledge.

## 8.2 Methods and mining results

We apply our measure to compare between the results of ten different exploratory data mining methods for binary data. Table 1 gives an overview, here we list the methods, stating the parameters we use, and giving how we refer to each of the methods between brackets.

*clus* We employ simple  $k$ -means clustering with  $k = 5$  clusters, using  $L_1$  distance. We turn the clusters into tiles by computing column margins inside each cluster.

*bicl* A bi-clustering simultaneously clusters the transactions and the items; a cluster is then a tile defined by the corresponding pair of item and transaction clusters (Pensa et al. 2005). We apply biclustering by separately clustering the columns and rows using again  $k$ -means clustering ( $k = 5$ ) and combine the two clusterings into a grid. Puolamäki et al. (2008) showed that a good approximation bound can be achieved with this approach. Each cluster is represented by a single tile.

*atcl* We use the parameter-free attribute clustering approach by Mampaey and Vreeken (2012, 2010) that groups together binary or categorical attributes

<sup>1</sup> The full version of the mammal dataset is available for research purposes upon request from the Societas Europaea Mammalogica. <http://www.european-mammals.org>.

<sup>2</sup> NOW public release 030717 available from (Fortelius et al. 2006).

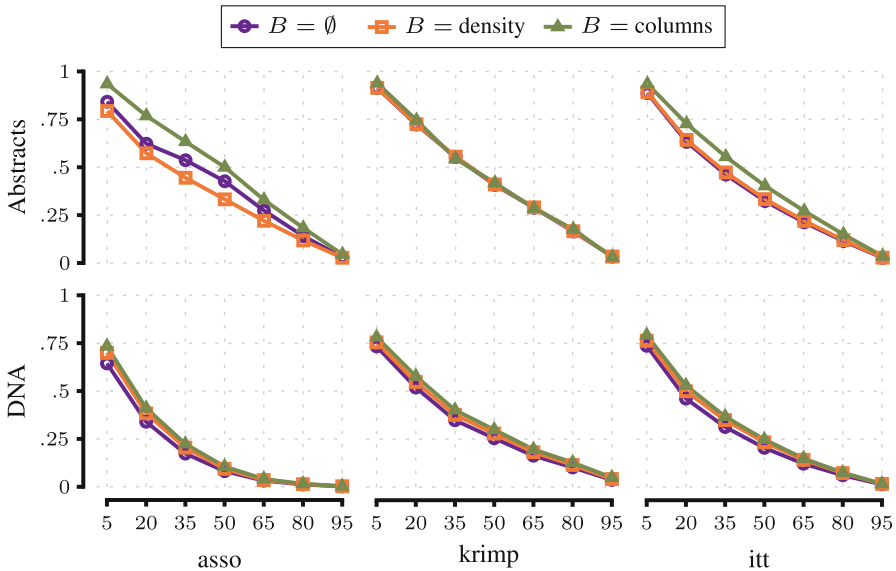
<sup>3</sup> <http://www.adrem.ua.ac.be/implementations/>.

- for which the values show strong interaction. We convert each attribute cluster into a tile—thereby somewhat oversimplifying these results, as we disregard the identified structure in attribute-value combinations within the cluster.
- sscl* Subspace clustering aims at finding groups of rows that exhibit strong similarity on a subset, i.e. subspace, of the attributes. For subspace clustering, we used the implementation of Müller et al. (2009) of the PROCLUS algorithm (Aggarwal et al. 1999), mined 100 clusters, each over maximally 32 dimensions, and converted each into a noisy tile.
- asso* The ASSO algorithm (Miettinen et al. 2008) approximates the optimal  $k$ -rank Boolean Matrix Factorisation of the data. We ran with a maximum of 100 factors, of which the non-empty ones were converted into tiles.
- tiling* Using the TILING algorithm (Geerts et al. 2004) we mine overlapping tilings, of up to 100 exact tiles, allowing the algorithm a maximum running time of 8 h.
- hyper* The HYPER algorithm is strongly related to Tiling, but can find noisy tiles by combining exact tiles such that error is minimised. Per dataset, we mined up to 100 hyper rectangles (Xiang et al. 2011), directly obtaining noisy tiles.
- itt* We mined Information-theoretic exact tiles (De Bie 2011b), where the method automatically selects the number of tiles, and ranks them according to informativeness. Out of the returned tiles, we use the top-100.
- mtv* We applied the MTV algorithm for obtaining the most informative itemsets; those itemsets for which the predicted frequency is most off given the current model (Mampaey et al. 2012). We allow a maximum run time of 2 hours, and a maximum number of 100 returned itemsets. As this algorithm focuses on frequency, selecting both itemsets for which the frequency is surprisingly high as well as surprisingly low, we convert these itemsets into tiles by selecting all rows.
- krimp* We used KRIMP (Vreeken et al. 2011) to mine itemsets that compress. From the resulting code tables, we identified the top-100 itemsets that are most used during compression, i.e. that aid compression best, and convert these into tiles by using their KRIMP-usage as *tid*-sets.

The last four of these methods select itemsets from a candidate collection. As candidate itemsets to select from, we used closed frequent itemsets mined at as low as feasible support thresholds, of resp. 5, 1, 800, and 1. For KRIMP, however, by its more optimised implementation we can use lower support thresholds for the *Abstract* and *Mammals* datasets, of resp. 4 and 300.

### 8.3 Measuring distances

First, we evaluate whether the measured distance converges to 0 when two sets of tiles approximate each other with regard to background knowledge. To this end, we take the tile sets of *asso*, *krimp*, and *itt*, as obtained on resp. the *Abstracts* and *DNA* datasets. In Fig. 2 we plot, per method, the measured distance between the top- $k$  and top-100 tiles. We give the measurements for three different background knowledge settings, resp. no background knowledge, knowledge of the average density of the dataset, and



**Fig. 2** Distance between top- $k$  tile sets and top-100 tile sets as a function of  $k$ . Rows represent datasets while the columns represent the methods

the column margins. The tiles are sorted according to their output order, for *asso* and *itt*, and ascending on code length for *krimp*.

As Fig. 2 shows, measurements indeed converge to 0 for higher  $k$ , i.e. when the two tile sets become more identical. Adding background information typically increases the distance. This is due to two reasons. First, when density is used, then we can infer additional differences between the areas that are not covered by tiles, thus highlighting the differences. Second, when we are using column margins, we reduce  $KL(\mathcal{M} \parallel \mathcal{B})$ , the joint information w.r.t. the background knowledge, consequently increasing the distance. Interestingly, for *Abstracts*, the distances for *asso* in fact decrease when density is used as background knowledge. This is caused by the fact that *asso* produces many overlapping tiles and these overlaps are emphasised when density is used.

### 8.4 Distances between results

Our main experiment is to investigate how well we can compare between results of different methods. To do so, for every dataset, and every method considered, we convert their results into tile sets as described above. We measure the pair-wise difference between each of these tile sets, using resp. the empty set, overall density, the column margins, and the combination of row and column margins, as background knowledge. For analysis, we present these numerical results, and the averages over the datasets, visually in Fig. 3 by plotting all pair-wise distances by Sammon projection (Sammon 1969). Sammon projection is a commonly used variant of Multi-dimensional scaling (MDS), in which one projects a high-dimensional space (here the pair-wise distance tables) onto a two-dimensional space, while preserving the structure of the individual



**Fig. 3** Sammon projections (a type of MDS, see Sammon 1969) of distances between tile sets. Each row represents a dataset and each column represents used background knowledge. Note that *atcl* and *mtv* are not included in the rightmost columns, as their tiles provide no information beyond column margins

distances as well as possible. We colour tiling and mark and clustering methods differently. Note that by our conversion into noisy tiles, we here treat the results of MTV (itemsets and their frequencies) as attribute clusters.

Considering the first column first, we see that without background knowledge three of the clustering approaches provide virtually the same result. We also see that, albeit not identical, the results of *asso*, *itt*, *krimp*, and *tiling* are relatively close to each other; which makes sense from a conceptual point of view, as these methods are methodologically relatively similar. For *DNA*, the measured dissimilarity between these methods lies between 0.28 and 0.38, whereas the dissimilarities to the other methods measure approximately 0.9.

We observe that *hyper*, while conceptually similar, is measured to provide different results when no background knowledge is given. This is mostly due to it including a few very large tiles, that practically cover the whole data, whereas the other methods

**Table 2** Redescribing the results of clustering (*clus*) by the results of resp. *asso*, *tiling*, *hyper*, *itt*, and *krimp*, using the FRUITS algorithm

Dataset	Redescription/# of tiles/full tile set														
	<i>asso</i>		<i>tiling</i>		<i>hyper</i>		<i>itt</i>		<i>krimp</i>						
Abstracts	.76	(70)	.76	.74	(38)	.74	.50	(32)	.57	.68	(100)	.68	.85	(98)	.85
DNA	.65	(11)	.83	.70	(9)	.79	.67	(21)	.84	.67	(11)	.81	.67	(19)	.81
Mammals	.30	(51)	.31	.68	(3)	.68	.34	(24)	.39	.78	(2)	.78	.49	(91)	.49
Paleo	.68	(16)	.83	.69	(28)	.78	.68	(23)	.81	.73	(21)	.81	.74	(38)	.79

Shown are, per combination, the dissimilarity between *clus* and the discovered redescription, the number of tiles of the redescription, and the dissimilarity measurement to the complete tile set

only cover the data partially. For *hyper* we see that once background knowledge is included, these large tiles are explained away, and the method subsequently becomes part of the ‘tiling’ group.

Clustering algorithms are close to each other when no background information is used because by covering all the data, they convey well that these datasets are sparse. When we use density as background knowledge, the differences between clusterings become visible. Interestingly enough, adding row margins to column margins as background information has small impact on the distances.

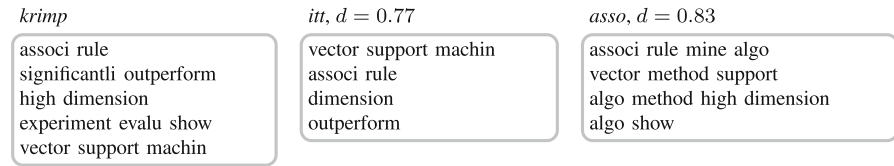
## 8.5 Redescribing results

Next, we empirically evaluate how our measure can be employed with regard to redescription results; both as validation as well as possible application.

To this end, we first investigate redescription results of completely different methods. As such, we take clustering as the target and density as the background information, and redscribe its result by using the tile sets of five of the pattern mining methods as candidate tile sets. Table 2 shows the results of these experiments: for four datasets, the measured divergence between the redescription and the target, the divergence of the complete tile set to the target, and the number of tiles selected for the redescription. First, and foremost, we see that redescription decreases the measured divergence, which correctly shows that by filtering out, e.g. too specific, tiles that provide information not in the target, we obtain a better description of the target.

We also see, with the exception of *Mammals*, that the measurements are quite high overall, suggesting the clustering provides information these results do not; not surprising, as these pattern mining methods focus on covering 1s, and not necessarily cover the whole data. Indeed, we see that by providing large and noisy tiles, and so covering more of the data, *asso* and *hyper* lead to the best redescriptions of the clustering results. In particular for *Mammals*, the target can be approximated very well, by resp. only half and a quarter of the total tiles. Overall, we note that typically only fractions of the full tile sets are selected, yet the amount of shared information is larger than for the full tile set: the pattern mining methods provide detailed local information not captured by clustering.





**Fig. 4** Redescribing, using the FRUTTS algorithm, (*left*) 5 tiles selected from the *krimp* result on the *Abstracts* dataset, by resp. tiles from (*centre*) *itt* and (*right*) *asso*

Second, we take a closer look at individual redescrptions. In order to be able to interpret these, we use the *Abstracts* dataset. We use *asso*, *krimp*, and *itt*, as these provide sufficiently many tiles to choose from; we leave *hyper* out, as for this data it mostly gives only very general tiles, covering all 1s in only 100 tiles.

By hand, we select 5 out of 100 *krimp* tiles, and we identify, for *asso* and *itt*, the sets of tiles that best approximate that partial result, and investigate how well the target concepts are approximated. In Fig. 4, we give an example. By the high distances, 0.77 and 0.83, we see the target is not approximated in detail. Overall, for *itt* we find only high-level translations that leave out detail, as its full tile set consists mostly of small itemsets. For *asso*, we see the redescription consists of target tiles combined with general concepts, which together give a reasonable approximation of the target. It is important to note that not simply all intersecting itemsets are given, but only those that provide sufficient information on the target; for both methods, overly large and overly general tiles (e.g. ‘high’) are not included in the redescription.

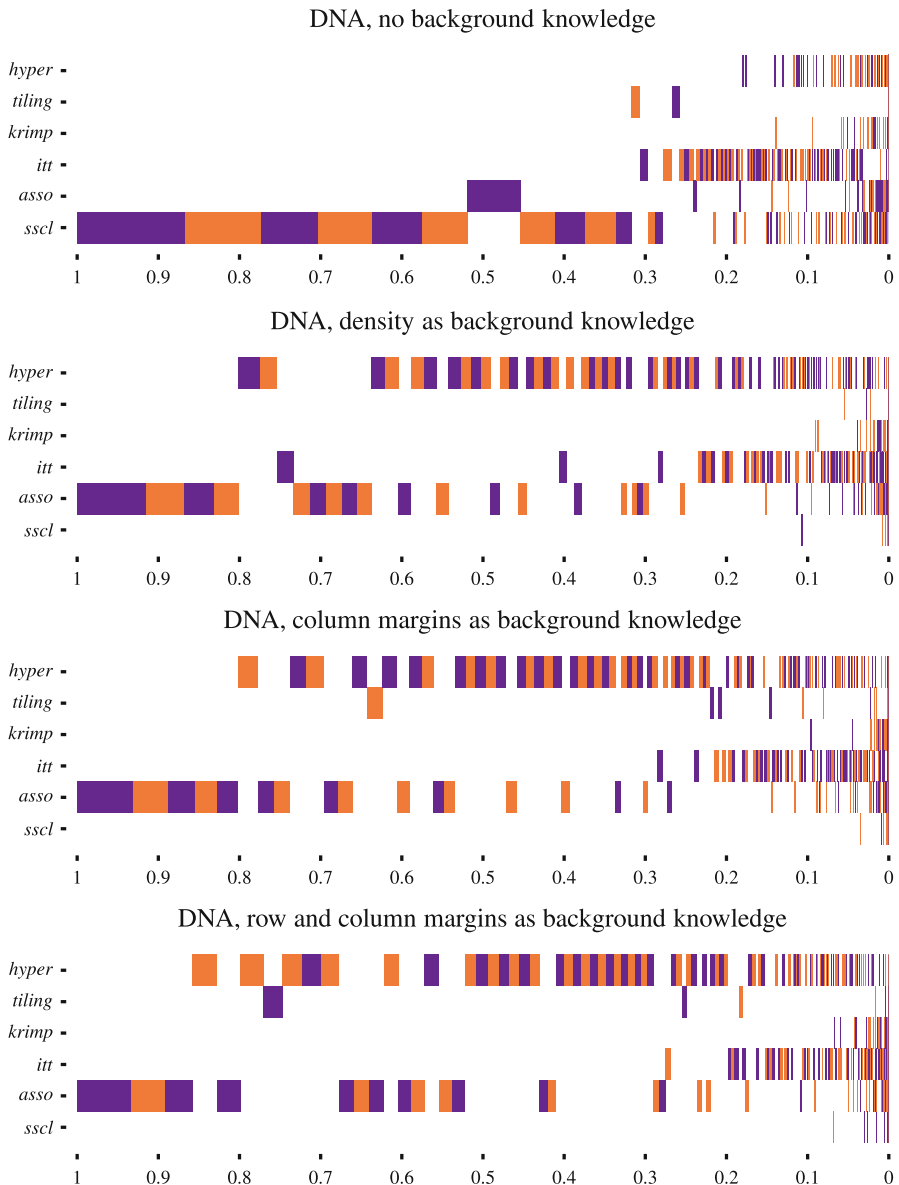
## 8.6 Iterative data mining

Next, we empirically evaluate how our measure can be employed for iterative data mining; iteratively identifying that (partial) result that provides most novel information with regard to the data.

First, for the *DNA* dataset, we combine the results of *hyper*, *tiling*, *krimp*, *itt*, *asso*, and *sscl* into one target tile set. Next, we apply the FITAMIN algorithm to rank these tiles by iteratively finding the tile that makes for the most informative addition to our selection. We do this for our four settings of background knowledge, and visually depict the results in Fig. 5.

We see that if we know nothing about the data, the *sscl* tiles, which cover relatively large areas of 0s, provide much information; whereas the tiles of the other methods, depicting mainly structure of the 1s, are ranked much lower. Once the density of the data is included in  $\mathcal{B}$ , we already know the data is relatively sparse—and subsequently tiles that identify relatively large areas of relatively many ones are initially most informative. For all non-empty background knowledge, we see the tiles of *tiling*, *krimp*, *itt*, and *sscl* to be presented quite late in the process. This makes sense, as these methods are aimed at identifying interesting local structure, which in an iterative setting is only informative once we know the bigger picture.

To reduce computation, FITAMIN employs a heuristic to determine which tile to add next to its selection—as opposed to iteratively fitting maximum entropy models for



**Fig. 5** Selection order (from left to right), and gain in information (bar width) with regard to the target of the combined results of *hyper*, *tiling*, *krimp*, *itt*, *asso*, and *sscl*, for the DNA. Top to bottom for background knowledge consisting resp. of the empty set, the data density, the column margins, and both column and row margins. The  $x$ -axis is the distance between the selected tiles so far and the full tile collection

every candidate, and measuring differences between those. To investigate the quality of this heuristic, we ran a variant of FITAMIN that does calculate exact distances for the *Paleo* dataset. While this variant is able to optimally locally select the best addition,

we only observe very slight differences in the final rankings; the main differences occur when we have the choice between two or more tiles approximately providing the same gain in information, and the tie is broken differently between the exact and heuristic approach. When overall we take a look at the individual differences between estimates and real distances, we see the estimates are of very high quality; the largest absolute recorded difference we recorded was only 0.007 (on a range of 0–1), which is close to numerical stability.

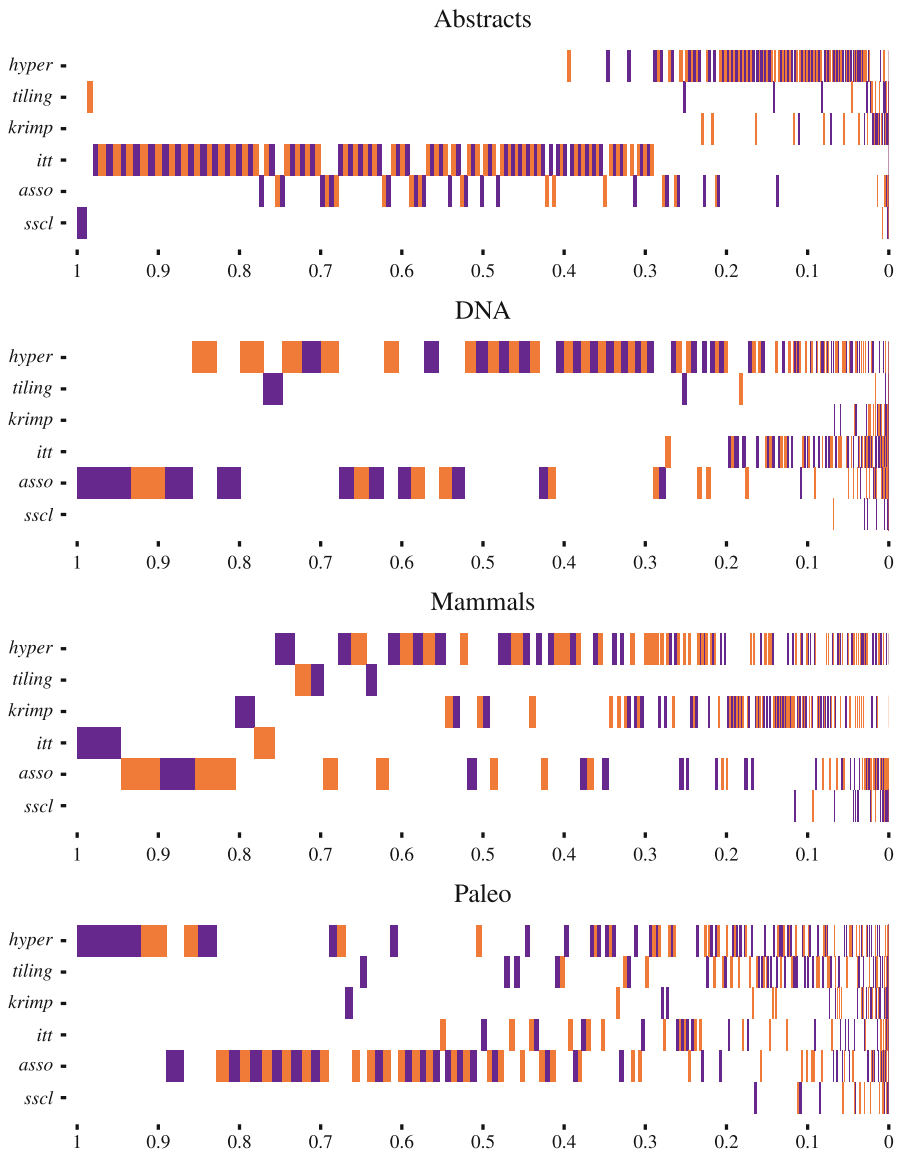
Next, for all datasets, using row and column margins as background knowledge, we show the selection order and gains in information as Fig. 6. As one would expect for datasets with different characteristics, different methods are best suited to extract the most information—there is no free lunch. Overall, and as expected, we observe that initially relatively large tiles not already explained away by the background knowledge are selected. As such, tiles from *asso* and *hyper* are often selected first, as these give a broad-strokes view of the data. Tiles that give much more local detail, or focus only on the 1s of the data, such as those mined by *krimp* and *itt* are most informative later on. Do note that FITAMIN does not give a qualitative ranking of results: our measure says nothing about interpretability, only about relative informativeness.

## 9 Discussion

The goal of this paper is to take a first step towards comparing between the results of different data mining methods. The method we propose here is for results obtained on binary data. By developing further maximum entropy modelling techniques, however, the same basic idea could be applied to richer data types. A recent result by [Kontonasios et al. \(2011\)](#) formalises a maximum entropy model for real-valued data. If this model can be extended to handle real-valued tiles, then our approach is also applicable for results on real-valued data. As for real-valued data statistics much richer than an average over a tile can be identified, a future work will involve in defining sufficiently rich theory formalising maximum entropy models that can take such statistics within tiles, such as correlations between attributes, into account as background information.

Besides measuring divergence of results between different methods, using the FRUITS algorithm we can identify high-quality redescriptions of results, and by FITAMIN we can heuristically approximate the optimal ranking of partial results based on their iterative informativeness. While beyond the scope of this paper, the latter algorithm can also be used rank complete results, i.e. tile sets. Moreover, our measure could be extended for choose the most informative result out of many runs of a randomised method, or, to measure differences between results when varying parameters of a method. Another, likely more challenging, extension would be to identify a ‘centroid’ or ‘medioid’ result that summarises a given large collection of results well.

Currently we rather straightforwardly convert results into tile sets. While we so capture much of the information they provide, we might not capture all information present in a result. Ideally, we would be able to encode structure beyond simple tile densities, such that the information captured in a result can be maintained more precisely when converted into sets of tiles. A desirable extension would be to be able to specify for a tile which attribute-value combinations occur how often within it, which



**Fig. 6** Selection order (from left to right), and gain in informativeness (width of the bars) with regard to the target tile set of the combined results of *hyper*, *tiling*, *krimp*, *itt*, *asso*, and *sscl*, for, from top to bottom, *Abstracts*, *DNA*, *Mammals*, and *Paleo*. The *x*-axis is the distance between the selected tiles so far and the full tile collection

would allow for better conversion of the results of attribute clustering (Mampaey and Vreeken 2010), as well as allow us to take simple itemset frequencies into account. A second refinement for future work would be to be able to specify complex structures in the data captured by statistics such as Lazarus counts (Fortelius et al. 2006), or nestedness (Mannila and Terzi 2007). Such refinements require further theory on

maximum entropy modelling. More complex modelling aside, our general approach of comparing results by comparing how many possible datasets exhibit the identified structure remains the same.

Our distance is based on maximum entropy model which uses the available information—that is the given tiles—as efficiently as possible. For example, if the frequencies of tiles in the second set can be derived from the first set, then the distance will be 0 between these two sets. However, we do not take into account how complex are the derivations for the human and this might lead to some counterintuitive results. Another restriction of the maximum entropy model we currently employ, is that it cannot recognise the informativeness of a data mining result identifying that the data is uniformly distributed. While this can be a highly informative, and powerful result, as our maximum entropy model essentially is the generalised uniform distribution under constraints, the uniform property follows naturally, is no deviation, and hence is not deemed informative. These restrictions are both due to our choice of modelling the space of all datasets by maximum entropy. While a well-founded choice, with many desirable properties, it is a choice nevertheless and future work may identify other probabilistic models that circumvent the above mentioned restrictions.

It is very important to note that our method is *not* a quality measure on data mining results. It solely measures the information shared between two sets of tiles; it does not measure the subjective quality of results, nor does it say anything about the ease of analysis of a result. Note that this is a good thing, and our explicit goal. Ease of analysis is highly subjective, being dependent on the expert and the available means. By our measure, the expert can check how informative different results are, and given this information decide which result to analyse in detail. If two results of different complexity of analysis are measured to be approximately equally informative, the logical decision would be to analyse the more simple one; only investing the time and effort of considering the more complex result in detail if the measurement shows it will likely provide a lot of information not also available in the simple result.

## 10 Conclusion

In this paper we discussed comparing results of different explorative data mining algorithms. We argued that any mining result identifies some properties of the data, and that, in an abstract way, we can identify all datasets for which these properties hold. By incorporating these properties into a model using the Maximum Entropy principle, we can measure the shared amount of information by Kullback–Leibler divergence. The measure we construct this way is flexible, and naturally allows including background knowledge, such that differences in results can be measured from the perspective of what a user already knows.

As a first step towards comparing results in general, we formalised our approach for binary data, showed that we can convert results into tiles, and discussed how to incorporate these into a maximum entropy model. Our approach provides a means to study and tell differences between results of different data mining methods. For applying the measure, we gave the FRUITS algorithm for parameter-freely identifying which parts of a given set of results best redescribe a given (partial) result, and the FITAMIN algorithm for iteratively finding that result that provides maximal novel information

with regard to what we have learned so far. Experiments showed our measure gives meaningful results, correctly identifies methods that are similar in nature, automatically identifies sound redescriptions of results, and is highly applicable for iterative data mining.

**Acknowledgments** The authors wish to thank, in alphabetical order: Tijn De Bie for his information-theoretic noisy tile miner (De Bie 2011b); David Fuhry for his implementation of HYPER (Xiang et al. 2011); Matthijs van Leeuwen for major contributions to the implementation of KRIMP (Vreeken et al. 2011); Stijn Ligot for running experiments with PROCLUS (Aggarwal et al. 1999; Müller et al. 2009); Michael Mampaey for the implementations of Attribute Clustering (Mampaey and Vreeken 2010, 2012) as well as MTV (Mampaey et al. 2012); Pauli Miettinen for his implementation of ASSO (Miettinen 2008); Nikolaj Tatti and Jilles Vreeken are both supported by a Post-Doctoral Fellowship of the Research Foundation—Flanders (FWO).

### A Proofs for Theorems

*Proof of Theorem 2* Let  $p^*$  be the maximum entropy distribution. Define a distribution  $q$

$$q(D) = \prod_{i,j} p^*((i, j) = D(i, j)).$$

Lemma 1 now implies that  $q \in \mathcal{P}$  so Theorem 1 implies that  $\{D \mid p^*(D) = 0\} \subseteq \{D \mid q(D) = 0\}$ . Assume that  $p^*(D) > 0$ . Then  $p^*((i, j) = D(i, j)) > 0$  and consequently  $q(D) > 0$ . This implies that  $\{D \mid p^*(D) = 0\} = \{D \mid q(D) = 0\}$ .

Assume that  $p^*(D) > 0$ , then we can decompose the sums of  $fr(T; D)$  in the exponential form in Theorem 1  $p^*(D)$  can be written as a product of potentials,  $p^*(D) \propto \prod_{i,j} \phi_{i,j}(D(i, j))$ . If we normalize each term by  $\phi_{i,j}(1) + \phi_{i,j}(0)$ , we have that  $p^*(D) = \prod_{i,j} p^*((i, j) = D(i, j))$  for  $p^*(D) > 0$ . If on the other hand  $p^*D = 0$ , then also  $q(D) = 0$ , so by definition  $p^*((a, b) = D(a, b)) = 0$  for some  $(a, b)$ . This implies that  $p^*(D) = 0 = \prod_{i,j} p^*((i, j) = D(i, j))$ .

Let us show that the individual terms have the stated form. Fix  $(i, j)$  such that  $0 < p^*((i, j) = 1) < 1$ . Let  $D$  be a dataset and let  $D'$  be the dataset with flipped  $(i, j)$ th entry. Then  $q(D) > 0$  if and only if  $q(D') > 0$  and hence  $p^*(D) > 0$  if and only if  $p^*(D') > 0$ . Divide the dataset space

$$\mathcal{D}_v = \{D \in \mathcal{D} \mid D(i, j) = v, p^*(D) > 0\}.$$

The previous argument shows that for each  $D \in \mathcal{D}_1$ , the dataset  $D'$  with flipped entry  $D' \in \mathcal{D}_0$ . Theorem 1 implies that  $p^*(D)/p^*(D') = \exp\left(\sum_{T \in \mathcal{T}(i,j)} \lambda_T\right)$ . We have now

$$\frac{p^*((i, j) = 1)}{p^*((i, j) = 0)} = \frac{\sum_{D \in \mathcal{D}_1} p^*(D)}{\sum_{D \in \mathcal{D}_0} p^*(D)} = \frac{\sum_{D \in \mathcal{D}_1} p^*(D)}{\sum_{D \in \mathcal{D}_1} p^*(D')} = \exp \sum_{T \in \mathcal{T}(i,j)} \lambda_T.$$

The theorem follows by rearranging the terms. □

*Proof of Theorem 3* We will prove the theorem by showing that  $p_T^*$  satisfies the conditions in Theorem 2. Let  $\mathcal{P}$  be the distributions satisfying the frequencies. Let  $\mathcal{Z}$  be the collection of datasets for which  $p(D) = 0$  for every  $p \in \mathcal{P}$ . Set  $\mathcal{Z}' = \{D \mid p_T^*(D) = 0\}$ . Let  $q \in \mathcal{P}$  and let  $(i, j) \in s(T)$ . Corollary 1 implies that  $q((i, j) = 1) = \alpha_T$ . This implies that  $\mathcal{Z}' \subseteq \mathcal{Z}$  and since  $p_T^* \in \mathcal{P}$ , we must have  $\mathcal{Z} = \mathcal{Z}'$ . By setting  $\lambda_T = 0$  for every tile we see that  $p_T^*$  has the exponential form given in Theorem 2. Hence,  $p_T^*$  is truly the maximum entropy distribution.  $\square$

Theorem 5 follows directly from the following lemma.

**Lemma 2** *Let  $\mathcal{T}$  and  $\mathcal{U}$  be two tile sets such that  $s(\mathcal{U}) \subseteq s(\mathcal{T})$ . Then  $KL(\mathcal{T} \parallel \mathcal{U}) = |s(\mathcal{T}) \setminus s(\mathcal{U})| \log 2$ .*

*Proof* Let  $X = s(\mathcal{T}) \setminus s(\mathcal{U})$ . Theorem 2 implies that

$$KL(\mathcal{T} \parallel \mathcal{U}) = \sum_{i,j} \sum_{v=0,1} p_T^*((i, j) = v) \log \frac{p_T^*((i, j) = v)}{p_U^*((i, j) = v)}.$$

The only non-zero terms in the sum are the entries  $(i, j) \in X$  for which  $p_T^*((i, j) = v) = 1$ . Hence we have

$$KL(\mathcal{T} \parallel \mathcal{U}) = \sum_{(i,j) \in X} 1 \times \log \frac{1}{1/2} = |X| \log 2.$$

This proves the lemma.  $\square$

To prove the next theorems we will need the following result.

**Lemma 3** *Let  $\mathcal{T}$  be a set of tiles. Let  $p^*$  be the corresponding maximum entropy model. Let  $q$  be a distribution such that  $fr(T; q) = fr(T; p^*)$  for  $T \in \mathcal{T}$ . Then*

$$\sum_{D \in \mathcal{D}} q(D) \log p^*(D) = \lambda_0 + \sum_{T \in \mathcal{T}} \lambda_T |s(T)| fr(T; p^*) = -H(p^*),$$

where  $\lambda_T$  are the weights as defined in Theorem 1, and  $\lambda_0$  is the normalisation constant of  $p^*$ .

*Proof* Let  $\mathcal{Z} = \{D \in \mathcal{D} \mid p^*(D) = 0\}$ . Note that  $q(D) = 0$  for any  $D \in \mathcal{Z}$ . Applying Theorem 1 leads us to

$$\begin{aligned} \sum_{D \in \mathcal{D}} q(D) \log p^*(D) &= \sum_{D \in \mathcal{D} \setminus \mathcal{Z}} q(D) \log p^*(D) \\ &= \sum_{D \in \mathcal{D} \setminus \mathcal{Z}} q(D) (\lambda_0 + \sum_{T \in \mathcal{T}} \lambda_T |s(T)| fr(T; D)) \\ &= \lambda_0 + \sum_{T \in \mathcal{T}} \lambda_T |s(T)| \sum_{D \in \mathcal{D} \setminus \mathcal{Z}} q(D) fr(T; D) \\ &= \lambda_0 + \sum_{T \in \mathcal{T}} \lambda_T |s(T)| fr(T; p^*). \end{aligned}$$



This proves the left side of the equation. To prove the right side apply the same argument with  $q = p^*$ . □

**Corollary 2** *Let  $\mathcal{A}$  and  $\mathcal{B}$  be two tile collections such that  $\mathcal{B} \subseteq \mathcal{A}$ . Then  $KL(\mathcal{A} \parallel \mathcal{B}) = H(\mathcal{B}) - H(\mathcal{A})$ .*

*Proof* Let  $p_{\mathcal{A}}^*$  and  $p_{\mathcal{B}}^*$  be two maximum entropy distributions. Since  $fr(T; p_{\mathcal{A}}^*) = fr(T; p_{\mathcal{B}}^*)$  for any  $T \in \mathcal{B}$ , we have

$$KL(\mathcal{A} \parallel \mathcal{B}) = - \sum_{D \in \mathcal{D}} p_{\mathcal{A}}^*(D) \log p_{\mathcal{B}}^*(D) - H(\mathcal{A}) = H(\mathcal{B}) - H(\mathcal{A}),$$

which proves the result. □

*Proof of Theorem 6* The case for exact tiles follows directly from Theorem 5. The general case follows from Corollary 2 by first noticing that

$$KL(\mathcal{M} \parallel \mathcal{U} \cup \mathcal{B}) = H(\mathcal{U} \cup \mathcal{B}) - H(\mathcal{M}) \leq H(\mathcal{B}) - H(\mathcal{M}) = KL(\mathcal{M} \parallel \mathcal{B})$$

and similarly for  $KL(\mathcal{M} \parallel \mathcal{T} \cup \mathcal{B})$ . Thus we have

$$d(\mathcal{T}, \mathcal{U}; \mathcal{B}) = \frac{KL(\mathcal{M} \parallel \mathcal{U} \cup \mathcal{B}) + KL(\mathcal{M} \parallel \mathcal{T} \cup \mathcal{B})}{KL(\mathcal{M} \parallel \mathcal{B})} \leq 2 \frac{KL(\mathcal{M} \parallel \mathcal{B})}{KL(\mathcal{M} \parallel \mathcal{B})} = 2,$$

which proves the result. □

*Proof of Theorem 7* Define  $\mathcal{X} = \mathcal{T} \cup \mathcal{B}$  and  $\mathcal{Y} = \mathcal{U} \cup \mathcal{B}$ . Let  $p_{\mathcal{X}}^*$ ,  $p_{\mathcal{Y}}^*$ ,  $p_{\mathcal{B}}^*$ , and  $p_{\mathcal{M}}^*$  be the corresponding maximum entropy distributions. Define a distribution  $q(D) \propto p_{\mathcal{X}}^*(D)p_{\mathcal{Y}}^*(D)$ .

We claim that  $q = p_{\mathcal{M}}^*$ . To prove this we first show that  $q$  satisfies the tiles in  $\mathcal{M}$ . Let  $e = (i, j)$  be an entry. Assume that  $e \in s(\mathcal{B})$ . Since the tiles in  $\mathcal{B}$  are exact, we have, due Corollary 1,

$$p_{\mathcal{X}}^*(e = 1) = p_{\mathcal{Y}}^*(e = 1) = p_{\mathcal{M}}^*(e = 1) = v, \quad \text{where } v = 0, 1.$$

Assume that  $v = 1$ . This means that  $p_{\mathcal{Y}}^*(D) = 0$ , and consequently  $q(D) = 0$ , if  $D(e) = 0$ . This implies that  $q(e = 1) = 1$ . Similar argument holds for  $v = 0$ . Hence,  $q$  satisfies tiles from  $\mathcal{B}$ .

Assume now that  $e \notin s(\mathcal{B})$ . Since  $q$  has an exponential form,  $q(e = 1)$  can be written in the form given in Theorem 2. Assume that  $e \in s(\mathcal{T})$ , then  $e \notin s(\mathcal{U})$ . Assume that  $0 < q(e = 1) < 1$ . This implies that the fraction given in Theorem 2 contains only weights from the tiles in  $\mathcal{T}$ . Hence,  $p_{\mathcal{X}}^*(e = 1) = q(e = 1)$ . If  $q(e = 1) = 0, 1$ , then either  $p_{\mathcal{X}}^*(e = 1) = 0, 1$  or  $p_{\mathcal{Y}}^*(e = 1) = 0, 1$ . Since  $e \notin s(\mathcal{Y})$ , we must have  $p_{\mathcal{Y}}^*(e = 1) = 1/2$ . Hence,  $p_{\mathcal{X}}^*(e = 1) = q(e = 1)$ . This implies that  $q$  satisfies tiles in  $\mathcal{T}$ . Similarly,  $q$  satisfies the tiles in  $\mathcal{U}$ . Hence  $q$  satisfies the tiles in  $\mathcal{M}$ .

Theorem 1 now implies that  $\{D \in \mathcal{D} \mid p_{\mathcal{M}}^*(D) = 0\} \subseteq \{D \in \mathcal{D} \mid q(D) = 0\}$ . To prove the other direction, let  $D$  be such that  $q(D) = 0$ . Then  $p_{\mathcal{X}}^*(D) = 0$

or  $p_{\mathcal{Y}}^*(D) = 0$ . Assume that  $p_{\mathcal{X}}^*(D) = 0$ . Theorem 1 implies that any distribution that satisfies  $\mathcal{X}$  must have vanish at  $D$ . Hence,  $p_{\mathcal{M}}^*(D) = 0$ . Consequently,  $\{D \in \mathcal{D} \mid p_{\mathcal{M}}^*(D) = 0\} = \{D \in \mathcal{D} \mid q(D) = 0\}$

Since  $q$  has the correct exponential form and satisfies the tiles in  $\mathcal{M}$ , Theorem 1 implies that  $q = p_{\mathcal{M}}^*$ .

We have the following properties

$$\begin{aligned} e \in s(\mathcal{B}) &\implies q(e = 1) = p_{\mathcal{B}}^*(e = 1) = p_{\mathcal{X}}^*(e = 1) = p_{\mathcal{Y}}^*(e = 1), \\ e \in s(\mathcal{T}) &\implies q(e = 1) = p_{\mathcal{X}}^*(e = 1) \quad \text{and} \quad p_{\mathcal{Y}}^*(e = 1) = p_{\mathcal{B}}^*(e = 1), \\ e \in s(\mathcal{U}) &\implies q(e = 1) = p_{\mathcal{Y}}^*(e = 1) \quad \text{and} \quad p_{\mathcal{X}}^*(e = 1) = p_{\mathcal{B}}^*(e = 1), \\ e \notin s(\mathcal{M}) &\implies q(e = 1) = p_{\mathcal{B}}^*(e = 1) = p_{\mathcal{X}}^*(e = 1) = p_{\mathcal{Y}}^*(e = 1). \end{aligned}$$

Theorem 4 now implies that

$$KL(\mathcal{M} \parallel \mathcal{B}) = KL(\mathcal{X} \parallel \mathcal{B}) + KL(\mathcal{Y} \parallel \mathcal{B})$$

and

$$KL(\mathcal{M} \parallel \mathcal{X}) = KL(\mathcal{Y} \parallel \mathcal{B}) \quad \text{and} \quad KL(\mathcal{M} \parallel \mathcal{Y}) = KL(\mathcal{X} \parallel \mathcal{B}),$$

which proves the theorem. □

*Proof of Theorem 8* Theorem 5 implies that we can prove the claim by showing that

$$1 - \frac{|A \cap B|}{|A \cup B|} \leq 1 - \frac{|A \cap C|}{|A \cup C|} + 1 - \frac{|B \cap C|}{|B \cup C|},$$

where  $A$ ,  $B$ , and  $C$  are finite sets. We can rewrite the inequality as

$$\frac{|A \cap C|}{|A \cup C|} + \frac{|B \cap C|}{|B \cup C|} - \frac{|A \cap B|}{|A \cup B|} \leq 1. \tag{2}$$

To prove the inequality we will modify the sets in such a way that it will only increase the left side.

Assume that there exists  $x \in C \setminus (A \cup B)$ . Removing  $x$  from  $C$  will decrease the denominators of the first two terms by 1, and thus increase the left side of the inequality. Hence we can assume safely that  $C \subseteq A \cup B$ .

Assume that there exists  $x \in (A \cap B) \setminus C$ . Adding  $x$  to  $C$  will increase the numerators of the first two terms by 1, and thus increase the left side of the inequality. Hence we can assume safely that  $A \cap B \subseteq C$ .

Now let us assume that we have three variables  $x \in A \cap B$ ,  $y \in A \setminus C$ , and  $z \in B \setminus C$ . If we remove  $x$  from  $A$  and  $B$  and  $C$ , add  $z$  to  $C$ , and add  $y$  to  $C$ , then the numerators and the denominators of the first two terms will increase by 1 and the numerator and denominator of the third term will decrease by 1. Since

$$\frac{x}{y} \leq \frac{x + 1}{y + 1},$$

for any  $0 \leq x \leq y \neq 0$ , it follows that the left side of Eq. 2 can only increase.

We repeat this replacement until two cases happen (this is possible since  $A, B$ , and  $C$  are finite). Either  $A \cap B = \emptyset$  or  $A \subseteq C$  (or  $B \subseteq C$ ).

Assume that  $A \cap B = \emptyset$ . Then Eq. 2 reduces to

$$\frac{|A \cap C|}{|A \cup C|} + \frac{|B \cap C|}{|B \cup C|} \leq 1.$$

Assume that there exists  $x \in A \setminus C$ . Then we can remove  $x$  from  $A$  and only increase the first term. Hence we can assume that  $A \subseteq C$  and similarly  $B \subseteq C$ . This implies that  $C = A \cup B$ , and we get

$$\frac{|A \cap C|}{|A \cup C|} + \frac{|B \cap C|}{|B \cup C|} = \frac{|A|}{|C|} + \frac{|B|}{|C|} = \frac{|A|}{|A| + |B|} + \frac{|B|}{|A| + |B|} = 1,$$

which proves the first case.

To prove the second case, assume that  $A \subseteq C$ . This implies that  $A \cup B \subseteq C \cup B \subseteq A \cup B \cup B = A \cup B$ , that is  $A \cup B = B \cup C$ .

Assume that there exists  $x \in B \setminus C$ . Removing  $x$  from  $B$  decreases the denominators of the second term and the third terms. We need to show that

$$\frac{|B \cap C|}{|B \cup C| - 1} - \frac{|A \cap B|}{|A \cup B| - 1} \geq \frac{|B \cap C|}{|B \cup C|} - \frac{|A \cap B|}{|A \cup B|}.$$

Replacing  $B \cup C$  with  $A \cup B$  we have

$$\frac{|B \cap C|}{|A \cup B| - 1} - \frac{|A \cap B|}{|A \cup B| - 1} \geq \frac{|B \cap C|}{|A \cup B|} - \frac{|A \cap B|}{|A \cup B|}.$$

Multiplying both sides with  $|A \cup B|(|A \cup B| - 1)$  leads us to

$$|B \cap C||A \cup B| - |A \cap B||A \cup B| \geq |B \cap C|(|A \cup B| - 1) - |A \cap B|(|A \cup B| - 1).$$

Eliminating the terms leads us to inequality  $|B \cap C| \geq |A \cap B|$  which is true because  $A \cap B \subseteq C$ . Thus removing  $x$  from  $B$  can only increase the left side, hence we can safely assume that  $B \subseteq C$ , that is  $C = A \cup B$ . This implies that  $A \cup C = B \cup C = A \cup B$ ,  $B \cap C = C$ , and  $B \cap C = B$ . But then

$$\frac{|A \cap C|}{|A \cup C|} + \frac{|B \cap C|}{|B \cup C|} - \frac{|A \cap B|}{|A \cup B|} = \frac{|A|}{|A \cup B|} + \frac{|B|}{|A \cup B|} - \frac{|A \cap B|}{|A \cup B|} = 1,$$

which proves the theorem. □

*Proof of Theorem 9* Let  $S = T \cup \{U\}$ . Corollary 2 implies that

$$KL(\mathcal{M} \parallel T \cup B) = H(T \cup B) - H(\mathcal{M}) \geq H(S \cup B) - H(\mathcal{M}) = KL(\mathcal{M} \parallel S \cup B).$$

Thus

$$\begin{aligned}
 d(\mathcal{S}, \mathcal{U}; \mathcal{B}) &= \frac{KL(\mathcal{M} \parallel \mathcal{U} \cup \mathcal{B}) + KL(\mathcal{M} \parallel \mathcal{S} \cup \mathcal{B})}{KL(\mathcal{M} \parallel \mathcal{B})} \\
 &\leq \frac{KL(\mathcal{M} \parallel \mathcal{U} \cup \mathcal{B}) + KL(\mathcal{M} \parallel \mathcal{T} \cup \mathcal{B})}{KL(\mathcal{M} \parallel \mathcal{B})} = d(\mathcal{T}, \mathcal{U}; \mathcal{B}),
 \end{aligned}$$

which proves the result. □

*Proof of Theorem 10* Since  $\mathcal{M} = \mathcal{T} \cup \mathcal{B}$ , the left side of the equation follows immediately. Write  $\mathcal{S} = \mathcal{U} \cup \mathcal{B}$ . To prove the right side apply Corollary 2 to obtain

$$\begin{aligned}
 \frac{KL(\mathcal{M} \parallel \mathcal{S})}{KL(\mathcal{M} \parallel \mathcal{B})} &= \frac{H(\mathcal{S}) - H(\mathcal{M})}{KL(\mathcal{M} \parallel \mathcal{B})} = \frac{H(\mathcal{S}) - H(\mathcal{B}) + H(\mathcal{B}) - H(\mathcal{M})}{KL(\mathcal{M} \parallel \mathcal{B})} \\
 &= 1 - \frac{KL(\mathcal{S} \parallel \mathcal{B})}{KL(\mathcal{M} \parallel \mathcal{B})},
 \end{aligned}$$

which proves the result. □

*Proof of Theorem 11* We prove the hardness by reducing EXACTCOVER, testing whether a cover has a mutually disjoint cover, to the decision version of REDESCRIBE.

Assume that we are given a ground set  $S$  with  $N$  elements and a cover  $\mathcal{C} = \{C_1, \dots, C_L\}$  for  $S$ . To define the tiles and the target frequencies we will first define a particular dataset  $D$ . This dataset has  $N + 1 + L$  transactions of length  $N$ , an item representing an item in the ground set  $S$ . The first  $L$  transactions correspond to the cover sets,  $(i, j)$  entry is equal to 1 if and only if  $C_i$  contains the  $j$ th item. The last  $N + 1$  transactions are full of ones.

The tile sets are as follows.  $\mathcal{B} = \emptyset$ .  $\mathcal{T}$  contains one tile with the last  $N + 1$  transactions and all items.  $\mathcal{U} = \{U_1, \dots, U_L\}$  contains  $L$  tiles such that  $a(U_i) = C_i$  and  $t(U_i) = \{i, L + 1, \dots, L + N + 1\}$ . Let  $\mathcal{V}$  be a subset of  $\mathcal{U}$ . Write  $f(\mathcal{V}) = \sum_{V \in \mathcal{V}} |a(V)|$ . Theorem 5 now implies that

$$d(\mathcal{V}, \mathcal{T}; \mathcal{B}) = \frac{(N + 1)(N - |a(\mathcal{V})|) + f(\mathcal{V})}{(N + 1)N + f(\mathcal{V})}.$$

If  $|a(\mathcal{V})| < N$ , then we can augment with  $V$  with a tile containing an item outside of  $a(\mathcal{V})$ . This will reduce the first term in the numerator at least by  $N + 1$  and increase the second term by  $N$ , at maximum. Thus such augmentation will decrease the distance and so we can safely assume that  $|a(\mathcal{V})| = N$ . This implies that  $d(\mathcal{V}, \mathcal{T}; \mathcal{B}) = \frac{f(\mathcal{V})}{(N + 1)N + f(\mathcal{V})}$ . This is a monotonic function of  $f(\mathcal{V})$ . Since  $a(\mathcal{V})$  contains all items,  $f(\mathcal{V}) \geq N$  and is equal to  $N$  if and only if the tiles in  $\mathcal{V}$  are disjoint. Hence EXACTCOVER has a solution if and only if, there is a redescription with distance equal to  $N / (N(N + 1) + N)$ . □

## References

- Aggarwal CC, Wolf JL, Yu PS, Procopiuc C, Park JS (1999) Fast algorithms for projected clustering. In: Proceedings of the ACM international conference on management of data (SIGMOD), Philadelphia, PA, ACM, pp 61–72
- Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th international conference on very large data bases (VLDB), Santiago de Chile, Chile, pp 487–499
- Cover TM, Thomas JA (2006) Elements of information theory. Wiley, New York
- Csiszár I (1975) I-divergence geometry of probability distributions and minimization problems. *Ann Probab* 3(1):146–158
- Darroch J, Ratcliff D (1972) Generalized iterative scaling for log-linear models. *Ann Math Stat* 43(5):1470–1480
- De Bie T (2011a) An information theoretic framework for data mining. In: Proceedings of the 17th ACM international conference on knowledge discovery and data mining (SIGKDD), San Diego, CA, ACM
- De Bie T (2011b) Maximum entropy models and subjective interestingness: an application to tiles in binary databases. *Data Min Knowl Discov* 23:1–40
- Fortelius M, Gionis A, Jernvall J, Mannila H (2006) Spectral ordering and biochronology of European fossil mammals. *Paleobiology* 32(2):206–214
- Gallo A, Miettinen P, Mannila H (2008) Finding subgroups having several descriptions: algorithms for re-description mining. In: Proceedings of the 8th SIAM international conference on data mining (SDM), Atlanta, GA
- Garriga GC, Junttila E, Mannila H (2011) Banded structure in binary matrices. *Knowl Inform Syst* 28(1):197–226
- Geerts F, Goethals B, Mielikäinen T (2004) Tiling databases. In: Proceedings of discovery science, pp 278–289
- Gionis A, Mannila H, Seppänen JK (2004) Geometric and combinatorial tiles in 0–1 data. In: Proceedings of the 8th European conference on principles and practice of knowledge discovery in databases (PKDD), Pisa, Italy, pp 173–184
- Hanhijärvi S, Ojala M, Vuokko N, Puolamäki K, Tatti N, Mannila H (2009) Tell me something I don't know: randomization strategies for iterative data mining. In: Proceedings of the 15th ACM international conference on knowledge discovery and data mining (SIGKDD), Paris, France, ACM, pp 379–388
- Hollmén J, Seppänen JK, Mannila H (2003) Mixture models and frequent sets: combining global and local methods for 0–1 data. In: Proceedings of the 3rd SIAM international conference on data mining (SDM), San Francisco, CA
- Jaynes E (1982) On the rationale of maximum-entropy methods. *Proc IEEE* 70(9):939–952
- Knobbe A, Ho E (2006) Pattern teams. In: Proceedings of the 10th European conference on principles and practice of knowledge discovery in databases (PKDD), Berlin, Germany, vol 4213. Springer, pp 577–584
- Kontonasiotis KN, De Bie T (2010) An information-theoretic approach to finding noisy tiles in binary databases. In: Proceedings of the 10th SIAM international conference on data mining (SDM), Columbus, OH, SIAM, pp 153–164
- Kontonasiotis KN, Vreeken J, De Bie T (2011) Maximum entropy modelling for assessing results on real-valued data. In: Proceedings of the 11th IEEE international conference on data mining (ICDM), Vancouver, Canada, ICDM
- Kuncheva LI, Whitaker CJ (2003) Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach Learn* 51(2):181–207
- MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Berkeley, Calif., 1965/66), vol I: Statistics. Univ. California Press, Berkeley, pp 281–297
- Mampaey M, Vreeken J (2010) Summarising data by clustering items. In: Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD), Barcelona, Spain. Springer, pp 321–336
- Mampaey M, Vreeken J (2012) Summarising categorical data by clustering attributes. *Data Min Knowl Discov* (in press)
- Mampaey M, Tatti N, Vreeken J (2012) Succinctly summarizing data with itemsets. *ACM Trans Knowl Discov Data* (in press)

- Mannila H, Terzi E (2007) Nestedness and segmented nestedness. In: Proceedings of the 13th ACM international conference on knowledge discovery and data mining (SIGKDD), San Jose, CA, ACM, p 489
- Miettinen P (2008) On the positive-negative partial set cover problem. *Inform Process Lett* 108(4):219–221
- Miettinen P, Mielikäinen T, Gionis A, Das G, Mannila H (2008) The discrete basis problem. *IEEE Trans Knowl Data Eng* 20(10):1348–1362
- Mitchell-Jones A, Amori G, Bogdanowicz W, Krystufek B, Reijnders PH, Spitzenberger F, Stubbe M, Thissen J, Vohralik V, Zima J (1999) *The Atlas of European Mammals*. Academic Press
- Müller E, Günemann S, Assent I, Seidl T (2009) Evaluating clustering in subspace projections of high dimensional data. In: Proceedings of the 35th international conference on very large databases (VLDB), Lyon, France, pp 1270–1281
- Myllykangas S, Himberg J, Böhling T, Nagy B, Hollmén J, Knuutila S (2006) DNA copy number amplification profiling of human neoplasms. *Oncogene* 25(55): 7324–7332
- Pensa RG, Robardet C, Boulicaut JF (2005) A bi-clustering framework for categorical data. In: Proceedings of the 9th European conference on principles and practice of knowledge discovery in databases (PKDD), Porto, Portugal, pp 643–650
- Puolamäki K, Hanhijärvi S, Garriga GC (2008) An approximation ratio for biclustering. *Inform Process Lett* 108(2):45–49
- Ramakrishnan N, Kumar D, Mishra B, Potts M, Helm RF (2004) Turning cartwheels: an alternating algorithm for mining redescrptions. In: Proceedings of the 10th ACM international conference on knowledge discovery and data mining (SIGKDD), Seattle, WA, pp 266–275
- Rasch G (1960) *Probabilistic Models for Some Intelligence and Attainment Tests*. Danmarks paedagogiske Institut
- Sammon J (1969) A nonlinear mapping for data structure analysis. *IEEE Trans Comput* 18:401–409
- Siebes A, Vreeken J, van Leeuwen M (2006) Item sets that compress. In: Proceedings of the 6th SIAM international conference on data mining (SDM), Bethesda, MD, SIAM, pp 393–404
- Tatti N (2006) Computational complexity of queries based on itemsets. *Inform Process Lett* 98(5):183–187. doi:[10.1016/j.ipl.2006.02.003](https://doi.org/10.1016/j.ipl.2006.02.003)
- Tatti N (2007) Distances between data sets based on summary statistics. *J Mach Learn Res* 8:131–154
- Tatti N, Vreeken J (2011) Comparing apples and oranges: measuring differences between data mining results. In: Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD), Athens, Greece. Springer, pp 398–413
- Vreeken J, van Leeuwen M, Siebes A (2007) Characterising the difference. In: Proceedings of the 13th ACM international conference on knowledge discovery and data mining (SIGKDD), San Jose, CA, pp 765–774
- Vreeken J, van Leeuwen M, Siebes A (2011) KRIMP: mining itemsets that compress. *Data Min Knowl Discov* 23(1):169–214
- Wang C, Parthasarathy S (2006) Summarizing itemset patterns using probabilistic models. In: Proceedings of the 12th ACM international conference on knowledge discovery and data mining (SIGKDD), Philadelphia, PA, pp 730–735
- Xiang Y, Jin R, Fuhry D, Dragan F (2011) Summarizing transactional databases with overlapped hyperrectangles. *Data Min Knowl Discov* 23(2):215–251
- Zaki MJ, Ramakrishnan N (2005) Reasoning about sets using redescription mining. In: Proceedings of the 11th ACM international conference on knowledge discovery and data mining (SIGKDD), Chicago, IL, ACM, pp 364–373. doi:[10.1145/1081870.1081912](https://doi.org/10.1145/1081870.1081912)